UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE BIOLOGIA VEGETAL

The IsomiR Window: the interface that bridges the complexity of miRNAs and their functional impact

Beatriz Viamonte de Sousa Ferreira

Mestrado em Bioinformática e Biologia Computacional
Especialização em Bioinformática

Dissertação orientada por:
Professor Doutor António Manuel Silva Ferreira
Doutora Andreia de Jesus Amaral Gomes Barbosa Fonseca

2017

Acknowledgements

First, I would like to thank my supervisors for all the time and patience dedicated to me. All your guidance was extremely important.

I also want to thank my boyfriend Diogo for supporting me at all times. Thank you for being some I can look after with admiration and to never make me think that quitting was even an option.

My parents have always supported me, and all their belief and strength truly helped me overcome this challenge.

I want to thank my friend Inês Viegas, for having accompanied me throughout my academic journey, and for always supporting me.

Thank you to all my friends, David, Nuno, Soraia, Luisa, Mariana A., Mariana G., João, Mameri, for all the patience.

Finally, I would like to thank the BioISI group for funding this project (Multi/04046).

Resumo

Os métodos de sequenciação de elevado débito, conhecidos como *Next-Generation Sequencing* (NGS), têm sido bastante usados nos últimos anos, permitindo obter, em paralelo, milhões de sequências de DNA ou de RNA. Estes métodos são muito aplicados no estudo de moléculas de RNA de pequenas dimensões, nas quais se incluem os microRNAs (miRNAs), sendo estes conhecidos como reguladores da expressão génica. Adicionalmente, estes métodos permitiram a descoberta de variantes dos miRNAs que exibem alterações na sua sequência. Estas variantes denominam-se isomiRs, podendo pertencer a três grupos: isomiRs 5', isomiRs 3', e isomiRs com mudanças internas.

Atualmente existem várias ferramentas de bioinformática que permitem a identificação sistemática de isomiRs. No entanto, apesar dos esforços destas ferramentas em fornecer plataformas computacionais especializadas para a análise de dados de sequenciação de RNAs de pequenas dimensões, estas têm em falta bastantes funcionalidades, não permitindo que o investigador receba todo o contexto dos dados, e, por consequência, a complexidade dos isomiRs na amostra não é devidamente explorada. Uma funcionalidade que está em falta nestas ferramentas é a possibilidade de o utilizador realizar de forma integrada a análise de anotação de sequências, incluindo a expressão diferencial, e a análise de impacto funcional dos isomiRs encontrados nas amostras. Outro aspeto importante é a maioria destas ferramentas não permitir analisar dados de NGS. As que permitem analisar estes dados, não permitem a análise em paralelo de vários ficheiros e apresentam limites de tamanho demasiado reduzidos para os ficheiros de dados NGS. Adicionalmente, muitas das ferramentas não disponibilizam uma interface gráfica, tornando a tarefa de analisar dados de sequenciação mais difícil para investigadores que não têm conhecimentos em bioinformática.

Desta forma, é importante a existência de uma ferramenta que integre todas as análises necessárias, nomeadamente a identificação de isomiRs num conjunto de dados, assim como a inferência do impacto funcional destas moléculas, e que possua uma interface gráfica fácil de usar. Assim, este projeto teve como objetivo contribuir para o desenvolvimento de uma ferramenta que permita a identificação rápida e eficiente de isomiRs e que integre diferentes funcionalidades de um modo automático, que vão desde a anotação de pequenos RNAs em dados de NGS à análise funcional para investigar o impacto biológico dos isomiRs identificados.

Como contribuição principal deste projeto foi criada uma aplicação *web*, que integra uma *pipeline* de bioinformática (fora do âmbito desta tese), e que suporta dois módulos de análise, de anotação e funcional, tendo sido considerada de raiz a transferência de informação entre os dois módulos de análise. Esta aplicação tem um conjunto mais completo de funcionalidades do que outras ferramentas existentes, apenas precisando de um *browser web* para poder ser usada. O funcionamento da aplicação foi testado utilizando dados de NGS disponíveis publicamente, tendo demonstrado a capacidade desta para processar vários ficheiros de uma forma integrada, produzindo gráficos e tabelas que demonstram os resultados deste processamento. Estes revelam uma complexidade das moléculas de pequenos RNAs não codificantes que não tinha sido previamente observada. Finalmente, foi criada uma máquina virtual com a aplicação desenvolvida, assim como todo o *software* da qual esta depende, de um modo pronto a usar, a qual está disponível no endereço http://isomir.fc.ul.pt.

Palavras-chave: IsomiRs; Análise de Anotação; Análise Funcional; Bioinformática; Aplicação Web

Abstract

Next-Generation Sequencing (NGS) methods have been widely used over the past years, allowing researchers to obtain, in parallel, millions of DNA and RNA sequences. These methods are extensively applied in the study of small RNA molecules, in which microRNAs (miRNAs) are included, which are known to act as regulators of gene expression. Additionally, NGS methods have permitted the discovery of variants of miRNAs, which exhibit changes in their sequence when compared to the canonical miRNA, and are called isomiRs. These molecules belong to one of three groups: 5' isomiRs, 3' isomiRs, and isomiRs with internal editings.

Nowadays, there are several bioinformatics tools that allow the systematic identification of isomiRs. However, they lack several key functionalities that prevent the user from understanding the entire complexity within the data, and consequently, the complexity of the isomiRs is not fully explored. One functionality that is absent in these tools, is an integrated workflow to sequentially, annotate sequences, infer differential expression, and assess the functional impact of isomiRs. Importantly, many of these tools do not accept NGS data as input. Regarding the ones that accept NGS data, they do not allow the analysis of several files in parallel and limit the size of the input in a way that excludes many NGS files. Furthermore, the lack of a graphical interface in these tools is also common, making the task of analyzing NGS data harder for researchers that are not familiar with bioinformatics concepts.

Thus, it is important to have a tool that integrates all the required analysis for isomiR identification and for inferring the functional impacts of those molecules, and that provides an easy to use graphical interface. Therefore, the main goal of this project was the development of a tool that allows a quick and efficient identification of isomiRs and that integrates different functionalities automatically, including the annotation of small non-coding RNAs in NGS data and the functional analysis so that the researcher can investigate the biological impact of the identified isomiRs.

The main contribution of this project was the development of a web application, which integrates a bioinformatics pipeline (outside the scope of this thesis), that allows the execution of two types of analyses, annotation and functional, having been built from scratch to support the sharing of data between the two analyses. This application presents a more complete set of functionalities, compared to other existing tools, and is available to the user through a web browser. The tool benchmarking was performed using publicly available NGS data, showing the ability to process multiple datasets in an integrated manner and producing reports of results in charts and table displays. These results show the complexity of small non-coding RNAs that had not been explored in the study. A virtual machine was created, in which the web application and pipeline are installed and configured as well as third-party software dependencies. The virtual machine is ready to use and it is available at http://isomir.fc.ul.pt.

Keywords: IsomiRs; Annotation Analysis; Functional Analysis; Bioinformatics; Web Application

Resumo alargado

Os métodos de sequenciação de elevado débito, conhecidos como *Next-Generation Sequencing* (NGS), têm sido bastante usados nos últimos anos, permitindo obter, em paralelo, milhões de sequências de DNA ou de RNA. Estes métodos produzem uma grande quantidade de informação num curto espaço de tempo, permitindo a sequenciação, de um modo rápido e de baixo custo, de um genoma ou de um transcriptoma.

Estes métodos são também muito aplicados no estudo de moléculas de RNA de pequenas dimensões (menores que 30 nucleótidos), nas quais se incluem os microRNAs (miRNAs). Os miRNAs ligam-se maioritariamente aos 3'UTR (*three prime untranslated region*) dos RNAs mensageiros, modelando a sua expressão, resultando na regulação da expressão génica por parte dos miRNAs. Adicionalmente, estes métodos permitiram a descoberta de variantes dos miRNAs que exibem alterações na sua sequência, e que se denominam isomiRs. Estes podem ser classificados em três grupos: isomiRs 5' (clivagem de nucleótidos na região 5' do miRNA), isomiRs 3' (clivagem ou adição de nucleótidos na região 3' do miRNA) e isomiRs com mudanças internas (substituição de nucleótidos na região interna do miRNA). Os isomiRs 5', devido ao facto de possuírem alterações na sequência da *seed*, têm um grande efeito na alteração da expressão génica da célula. No que toca aos isomiRs 3', apesar de não implicarem a existência de alterações na região *seed*, foi demonstrado que também causam impacto na regulação do programa de expressão génica em comparação com o respetivo miRNA canónico. Os isomiRs com mudanças internas, alterando a sequência nucleotídica, também podem ter afinidades de ligação para diferentes genes alvo em comparação com o miRNA canónico.

Atualmente existem várias ferramentas de bioinformática que permitem a identificação sistemática de isomiRs. No entanto, apesar dos esforços destas ferramentas em fornecer plataformas computacionais especializadas para a análise de dados de sequenciação de RNAs de pequenas dimensões, estas têm em falta bastantes funcionalidades, não permitindo ao investigador obter uma caraterização completa da complexidade dos pequenos RNAs presentes numa amostra.

Uma funcionalidade em falta nas ferramentas para a análise de dados de sequenciação de RNAs de pequena dimensão, é a possibilidade do utilizador poder, de um modo integrado, identificar todos os tipos de isomiRs e poder inferir qual o impacto destes no contexto experimental em estudo. Deste modo é necessário integrar a análise de anotação (que consiste na anotação de sequências) e que permite revelar a complexidade de pequenos RNAs não codificantes expressos (nomeadamente miRNAs, isomiRs e outros RNAs não codificantes); a análise de expressão diferencial, que permite identificar quais os miRNAs e isomiRs que estão diferencialmente expressos; e a análise funcional, na qual se efetua a previsão dos alvos dos isomiRs e se infere quais os processos biológicos mais afetados pelos isomiRs detetados. A falta de um fluxo de trabalho que integre todas estas funcionalidades obriga o investigador a realizar estas análises através do uso de múltiplas ferramentas, o que pode ser uma tarefa exaustiva e que, inclusivamente, está sujeita a mais erros.

A maioria das ferramentas disponíveis para a análise de isomiRs não disponibiliza uma interface gráfica, o que torna a tarefa de analisar dados de sequenciação mais difícil para investigadores que não tenham conhecimentos de bioinformática. Muitas das ferramentas existentes exigem ao utilizador conhecimentos de Linux e requerem a instalação de *software* através de linha de comandos, além da manipulação de ficheiros com milhões de dados, o que é uma barreira à utilização da ferramenta pela maioria dos potenciais utilizadores, como biólogos, investigadores biomédicos e clínicos. Atualmente, e que seja do nosso conhecimento, apenas duas ferramentas, CPSS e DeAnnIso, apresentam uma interface gráfica que permite a realização integrada das análises de anotação e de impacto funcional. No entanto, estas apresentam restrições que não permitem a análise expedita de dados NGS, nomeadamente devido ao limite de 50 MBytes para o ficheiro de dados NGS, que é bastante inferior ao tamanho habitual

de, pelo menos, 1 GByte. Acrescenta-se que resultados apresentados por estas ferramentas são, em alguns casos, pouco objetivos e apresentam falta de informação relativa à complexidade dos pequenos RNAs não codificantes. Desta forma, é importante a existência de uma ferramenta que permita realizar, de modo integrado, todas as análises necessárias à caraterização de pequenos RNAs não codificantes. Esta deve centralizar toda a informação relativa a isomiRs presentes nas amostras de estudo, e também ter uma interface gráfica que seja fácil de usar.

Assim, este projeto teve como objetivo contribuir para o desenvolvimento de uma ferramenta (IsomiR Window) que permita a identificação rápida e eficiente de isomiRs e que integre diferentes funcionalidades de um modo automático, que vão desde a anotação de pequenos RNAs em dados de NGS à análise funcional para investigar o impacto biológico dos isomiRs identificados. Esta ferramenta permite a análise paralela de vários ficheiros de dados NGS e não tem restrições quanto ao tamanho dos ficheiros. Esta providencia a visualização de resultados que caraterizam a complexidade das moléculas de RNA detetadas, com qualidade adequada à posterior disseminação de resultados. A visualização de resultados deverá possibilitar ao utilizador um nível de interatividade de modo a que o resultado final esteja de acordo com os seus objetivos. Este projeto foi desenvolvido tendo como base uma aplicação *web*, que torna a análise dos dados de utilização fácil para investigadores com um nível de conhecimento e experiência na utilização de ferramentas de bioinformática bastante incipiente e que integra uma *pipeline* de bioinformática (fora do âmbito desta tese) que realiza o processamento de dados NGS.

A aplicação foi desenvolvida em várias fases. Na primeira fase foi efetuada uma revisão da literatura sobre a biologia dos isomiRs e das aplicações que permitem a análise destas moléculas, e também um estudo das metodologias e linguagens de programação para o desenvolvimento *web*. Na segunda fase foi feita a análise e *design* da aplicação *web*, que teve como um dos resultados um diagrama *User Environment Design*, que define o fluxo de trabalho e funcionalidades suportadas pela aplicação. Na terceira fase foi feita a implementação da aplicação, que está divida em três camadas, nomeadamente: camada de lógica aplicacional (*backend*); a camada de apresentação (*frontend*), e a camada do *pipeline*.

Relativamente ao *backend*, foi criada uma base de dados que guarda as informações relativas às análises de anotação e funcional e respetivas interações. Foram também desenvolvidos serviços *web*, com o protocolo REST (*Representational State Transfer*), que permitem obter informações sobre as análises e para comunicar com a camada do *pipeline*, por exemplo para dar início a uma nova etapa de processamento ou para reportar erros. O *frontend* é responsável por oferecer uma interface gráfica ao utilizador num *browser web*, com o qual acede às funcionalidades da aplicação. Relativamente ao *pipeline*, que consiste em vários *scripts* de Perl e R, este é usado para processar os dados inseridos pelo utilizador, tendo colaborado na definição de mensagens de erro e de finalização das etapas. As duas primeiras camadas foram implementadas utilizando a *framework* de PHP, Laravel. Na última fase do projeto foi feita uma avaliação da aplicação. Dados humanos de sequenciação foram submetidos na aplicação e foi feita a apresentação dos resultados relativos a estes dados.

Desta forma, como uma das contribuições principais deste projeto, foi criada uma aplicação *web* que permite a realização de dois módulos de análise, de anotação e funcional, tendo sido considerada de raiz a transferência de informação entre os dois módulos. A aplicação desenvolvida, IsomiR Window, tem um conjunto mais completo de funcionalidades do que outras ferramentas existentes, permitindo que um utilizador, através de um *web browser*, tenha acesso às mesmas através do preenchimento de simples formulários. Foi criada uma máquina virtual com a aplicação, assim como todo o *software* da qual esta depende, de um modo pronto a usar, a qual está disponível no endereço http://isomir.fc.ul.pt. A informação sobre os requerimentos da ferramenta, e também erros encontrados ao longo das análises, é fornecida ao utilizador em todos os contextos das análises.

O funcionamento da aplicação foi testado utilizando dados de NGS disponíveis publicamente, tendo revelado uma complexidade das moléculas de pequenos RNAs não codificantes que não tinha sido previamente observada.

Table of contents

List of figures

List of tables

Chapter 1 – Introduction

The IsomiR Window project aims to develop a user-friendly tool that enables biologists to perform the analysis of small-RNA-seq data, uncovering the complexity of miRNA biogenesis. The tool enables the discovery of all types of miRNA variants through a web interface that interacts with a Perl based pipeline. This thesis is focused on the development of a user-friendly analysis interface aiming to include enhanced visualization features, establishing a new bridge between small-RNA-seq data and the underlying biology of small non-coding RNAs with a clear focus on miRNAs.

This first chapter addresses the context and the motivation for the thesis, as well as its main goals. Additionally, the methodology applied and the document overview is also provided.

1.1 Context and motivation

Next-Generation Sequencing (NGS) has been greatly used in past years since it allows to sequence, in parallel, millions of sequencing reads, producing a great amount of data in a small period of time [1]. Specifically, small-RNA sequencing (small-RNA-seq), which uses NGS, allows to obtain the sequences of all types of small-non-coding RNAs (sncRNAs) existent in a sample, in which are included miRNAs, sncRNAs that are able to regulate gene expression [2]. Importantly small-RNA-seq data enabled to reveal the existence of frequent variations of miRNA canonical sequences, generating multiple variants: the isoforms of miRNAs (isomiRs) [2]–[6]. To extract the relevant biological information from this type of data, the development of bioinformatics tools that will enable developing integrative analysis in a user-friendly manner is in high demand.

Currently, that are several bioinformatics tools that allow the systematic identification of isomiRs [7]–[18]. However, despite the effort of these software tools to provide specialized computational frameworks to analyze small-RNA-seq data, they usually lack features, not allowing the researcher to capture the full complexity of isomiRs.

One feature that some of the tools lack is a complete analysis of small-RNA-seq data, in other words, they lack the possibility to perform annotation analysis and functional analysis in an integrative manner [7], [10], [12]–[14], [16], [19]. This means that in these cases, the researcher is forced to use multiple software tools to obtain the intended output. Additionally, in some of the tools that only provide annotation analysis, the results produced do not contain all the information the researcher needs. For example, the prediction of novel miRNAs, the detection of other non-coding RNAs, and the detection of miRNAs holding single nucleotide polymorphisms or editing events, are results that are not produced by the existing tools that perform annotation analysis [10], [12], [14], [16], [19], making them somewhat incomplete.

Another feature that the majority of these tools lack is a graphical-user interface (GUI) [9]–[15], [17], making them difficult to be used by researchers that do not have computational or bioinformatics knowledge. This can be very frustrating for the researcher because, without the proper knowledge, installing one or more technologies through the command line can become a difficult task to perform due to software dependencies and compatibility.

To our knowledge, only two tools, CPSS [8] and DeAnnIso [18], present a GUI and perform annotation and functional analysis. However, now focusing only on DeAnnIso (which is an updated version of CPSS), it only allows the analysis of one dataset at a time and displays restrictions in file size, which are not compatible with the expected size of NGS datasets (larger than 1Gb). Furthermore, the produced results do not allow capturing the complexity and the biological relevance of the identified

isomiRs in an objective and interactive manner. In other cases, such as the differential expression table, the results are too extensive, increasing the probability for user's frustration.

Therefore, this project aimed to contribute to the development of a tool (IsomiR Window) that allows the integrative analysis of small-RNA-seq data, allowing to process several datasets in parallel, without file size restriction and with a GUI that is quick and easy to use. This user-friendly GUI enables the analysis of NGS data, allowing the characterization of sncRNAs, with a special focus in isomiRs and integrating all relevant functionalities in the same application, without the need for the researcher to use several applications and perform manual processing. It is also intended to provide the interactive visualization of results, characterizing objectively the complexity of sncRNAs found in the data.

The user-friendly interface was developed based on the web architecture, which enables a user with little or no experience to perform NGS data analysis, providing a real bridge between biology and informatics. The web application integrates a bioinformatics pipeline (developed in parallel by a colleague and outside the scope of this thesis), enabling to query small-RNA-seq datasets for all possible types of isomiRs and further allowing the investigation of their functional consequences at the cell level. In Figure 1.1 are shown the two main components of the IsomiR Window tool, web application and pipeline, both sharing responsibilities for enabling annotation and functional analysis.

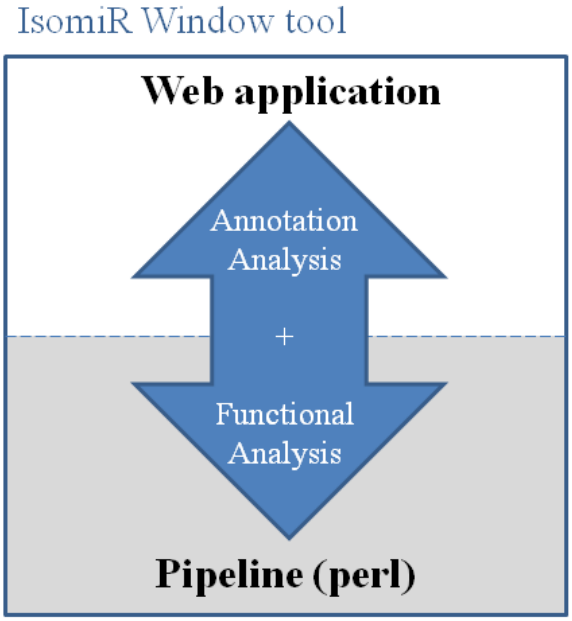


Figure 1.1. IsomiR Window tool components.

1.2 Goals

The first goal of this project was to develop a graphical, user-friendly, interface (GUI) for a web application, which is one of the main components of the IsomiR Window tool, allowing a comprehensive analysis of sncRNAs with a focus on isomiR species by processing small-RNA-seq datasets.

The second goal of this thesis was to test the developed tool using publicly available small-RNA-seq data. The selected datasets came from a study [20] aiming to profile changes in miRNA expression occurring in human naive CD4 T cells, in response to T cell receptor (TCR) stimulation and to infection with the human immunodeficiency virus type I (HIV1). However, in that study the miRNA complexity at the isomiR level was not explored, therefore this data was used to benchmark the IsomiR Window tool.

1.3 Methodology

To accomplish the proposed goals, the project was organized in four different stages, namely familiarization, system analysis and design, development of a backend for handling domain logic and a frontend for providing the user interface, and tool benchmarking.

The familiarization stage consisted in a review of the state of the art literature regarding isomiR biology and available methods for analysis. The comparison between the existing tools for isomiR analysis was also performed.

For the system analysis, a study of tool requirements was performed, and for the system design, a User Environment Design (UED) [21] diagram was built to define the workflow and functions supported by the application. When building the diagram, the connection between the two types of analysis, annotation and functional, within the application, was always considered.

In the development of the backend and frontend, several technologies were used, such as the Laravel framework and PHP language, the MySQL database, HTML and CSS, JavaScript and AJAX, and Highcharts (library used to develop charts). Web services were developed, through REST (Representational State Transfer) protocol, to obtain information about the analyses and to communicate with the pipeline layer, for example to initiate a new phase within an analysis or to report errors.

For the tool benchmarking, a set of small-RNA-seq datasets derived from human T cells was used.

1.4 Contributions

This thesis contributed to the IsomiR Window project at two different levels. The first contribution was the development of a web application, built from scratch that allows a comprehensive analysis of sncRNAs with a focus on isomiRs. This web application invokes a pipeline for data analysis allowing, through the GUI interface, the utilization by users with few or no bioinformatics skills. A virtual machine featuring a free operating system, the web application and pipeline, and all software packages they depend upon to run was also built. This virtual machine is ready to use through a virtualization software, such as VirtualBox. IsomiR Window, integrates different functionalities crucial for the identification of isomiRs and the understanding of their biological impact that are lacking currently available tools.

The second contribution was the benchmarking of IsomiR Window tool using human small-RNA-seq datasets. Through this benchmarking it was proven that the results produced by IsomiR Window allow exploring miRNA complexity in human datasets, which has not been previously observed.

1.5 Document overview

The document is divided into five chapters and it is organized in the following order:

- Chapter 1 consists in a summarized description of the entire project.
- Chapter 2 presents the key concepts for the development of the project, concerning bioinformatics and web development. In addition, a critical analysis of competing applications is also presented.
- Chapter 3 describes the different steps in the design and development of the web application.
- Chapter 4 describes the different parts of an analysis of real datasets in the developed web application. Descriptions of the data, results of the analysis, and conclusions that can be drawn from them, are presented.
- Chapter 5 presents the final conclusions of the project and some suggestions for future work.

Chapter 2 – Concepts and related work

The current chapter has been divided into three sections. The first section consists in the description of the state of the art concerning microRNA biology. The second section describes several tools that perform isomiR analysis. In the third and last section, the concepts concerning web development are described.

2.1 MiRNA biology and complexity in NGS data

MicroRNAs (miRNAs) are small non-coding RNAs (sncRNAs), with the approximate size of 22 nucleotides. These small non-coding RNAs act as post-transcriptional central regulators of gene expression by the induction of transcript degradation or translational inhibition of their target messenger-RNAs (mRNAs) [22], making them, also relevant disease biomarkers.

MiRNAs originate from double-stranded RNAs (dsRNAs), called primary miRNAs (pri-miRNAs) [23]–[25]. Pri-miRNAs are later processed into precursor miRNAs (pre-miRNAs) by the RNA endonucleases Drosha and Pasha [26]. After that, the pre-miRNAs are exported to the cytoplasm, where they are processed into a miRNA-miRNA duplex by two rounds of endoribonuclease cleavage, being Dicer the most relevant enzyme in this process [27]. Upon processing, one strand of the miRNA duplex (mature miRNA) will target mRNAs, mediating all RNA silencing pathways [28]. The miRNA biogenesis pathway is displayed in Figure 2.1 below.

The usage of high-throughput sequencing methods, also known as Next-Generation Sequencing (NGS), have the ability to sequence many different small RNAs at single-base resolution, on a genome-wide scale and in a single instrument run, at a very reduced cost when compared to traditional methods [1]. NGS has been broadly used to study the small RNA transcriptome [2], [29]. The only disadvantage of this technology it is the fact that each sequencing experiment produces up to 3 Gbp of sequence data, whose analysis represents an important bioinformatics challenge, even though a considerable number of user-friendly and efficient analysis software have been developed to fill the gaps.

2.1.1 MicroRNA variants: isomiRs

The use of NGS allowed researchers to observe modifications of the canonical miRNAs that comprise the addition or trimming of nucleotides at either end of the miRNA, and also nucleotide substitutions. All of these processes originate miRNA variants, that, when detected, became known as isomiRs [3]–[6], [30], [31].

IsomiRs can be classified into three different groups: 5’ isomiRs, 3’ isomiRs and internal isomiRs. The most predominant process that leads to the production of 5’ and 3’ isomiRs, is the imprecise and alternative cleavage, by Dicer or Drosha [4]. After this cleavage, the isomiRs generated will match the genomic template but differ (having a smaller number of nucleotides) in their 5’ and/or 3’ end positions. 5’ isomiRs, in particular, can also arise from differential processing of paralogous pre-miRNAs.

Furthermore, there are other mechanisms by which 5’ and 3’ isomiRs can be generated. One of these processes is nucleotide addition at the 3’end of a miRNA, and is most commonly characterized by uridylation or adenylation events. One or more nucleotidyl transferase enzymes catalyze these events. If the isomiRs with the added nucleotides align perfectly with the pre-miRNA, they are then defined as a ‘templated’ isomiRs, otherwise they are defined as ‘non-templated’ isomiRs [32], [33]**.** A special case of nucleotide addition is the addition of non-templated nucleotides by terminal nucleotidyl transferases, giving rise to a special form of 3´ isomiRs. This process is mainly known as tailing.

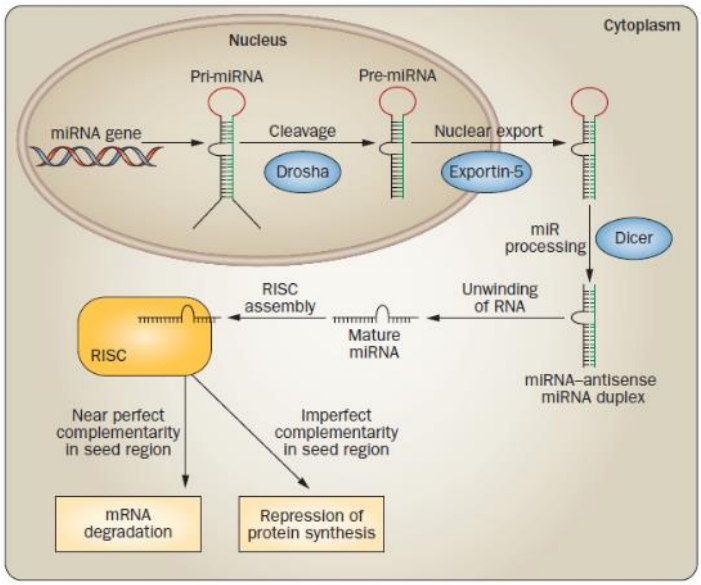


Figure 2.1. The miRNA processing pathway [34].

Another post-transcriptional modification is exonuclease-mediated nucleotide trimming, and this process is another mechanism for the production of 3' isomiRs [35]. The 3′ end trimming, that produces this type of isomiR, requires the 3′-to-5′ exoribonuclease Nibbler [36].

It is important to refer that some isomiRs undergo changes at both 5' and 3', causing them to be mixed isomiRs.

MiRNA editing is another process that leads to the generation of isomiRs by post-transcriptional enzymatic editing of the miRNA sequence [4], [5]. The most dominant type of miRNA editing is adenosine deamination, which is frequently referred to as adenosine-to-inosine (A-to-I) editing, and occurs with highest frequency in noncoding regions [37]. This process occurs at frequencies that are similar to sequencing errors [34], and they are often found in isomiRs that originate from miRNA family members from different loci [35]. These variations must be distinguished from nucleotide variations detected in isomiRs derived from the existence of single nucleotide polymorphisms (SNPs) in miRNA genes [38].

In general, in a given miRNA family containing variants, the miRNA and its isomiRs often display the same 5' end, while the 3' ends are significantly more variable and often affect the length of the isomiRs [39]. Importantly, while being less frequent 5′ end changes of the miRNA have a stronger functional impact, since this region harbors the miRNA seed region. The seed region is located in nucleotides 2–7 [28], [40], and corresponds to the canonical site of target recognition. The complementarity between a miRNA and its target can be perfect involving the full sequence of the miRNA or it can be partial, involving solely the miRNA seed region, reason why a single miRNA can regulate the expression of thousands of different targets. [40]. Importantly, the 5′ ends of miRNAs are constrained because argonaute protein loading typically selects for miRNAs with distinct 5′ nucleotides [41], [42].

Given that the regulatory roles of miRNAs are mainly the result of sequence specific binding to their target RNAs, it is possible to consider that isomiRs might function redundantly. However, recent studies showed that small differences in miRNA sequences are sufficient to give unique functions to isomiRs through differential target selection [43].

Furthermore, since both miRNA and its isomiRs originate from the same pre-miRNA, it is assumed that the expression patterns of isomiRs are analogous to those of their representative miRNA. However, it has been shown that some isomiRs accumulate in different conditions when compared to their

corresponding mature miRNAs [44], which could mean that the processing or modification of these miRNAs may be differentially regulated, and that their regulation of target RNAs is more dynamic.

In summary, 5' end isomiRs cause a major effect in animal systems because the miRNA seed region is located within 5' end of the miRNA. These isomiRs will typically bear a different seed sequence in comparison with the canonical miRNA, which is indicative of their potential to target different transcripts. Additionally, variations at the 3' end of isomiRs are mainly associated with different lengths of the miRNAs, but by changing the affinity between the miRNA and their targets may also impact their function. Finally, internal editing in miRNA family members may affect the binding affinity for a specific target. Since a miRNA frequently targets many genes in the same family, an isomiR may preferentially target transcripts that are not frequent targets of the canonical miRNA and the sum of their activities may result in the regulation of the expression of a gene family [30].

The upcoming sections approach the different types of analysis that can be performed to assess small-RNA-seq data, and more specifically, miRNAs and their isoforms present in that data.

2.1.2 Annotation and differential expression analysis

Annotation analysis allows a comprehensive analysis of the small RNA transcriptome entities contained in the studied dataset. More specifically, the annotation of sequences provides detailed information of each sncRNA found in each dataset, namely the detection of miRNAs, isomiRs and of other non-coding RNAs, along with the prediction of novel miRNAs. In summary, this analysis allows the researcher to investigate the complexity of sncRNA biology existent in the studied biological samples.

In addition to this analysis, the detection of differentially expressed miRNAs (including their isomiRs) is also very useful to the understanding of the impact of differential miRNA processing in the studied biological context.

Files used in annotation analysis

The file format normally used for this type of analysis is FASTQ and FASTA. The FASTQ file format uses four lines per sequence. This first line begins with a '@' character followed by a sequence identifier and an optional description. The second line corresponds to the sequence, the third line begins with a '+' character that can be followed by the sequence identifier. The final line contains the sequence quality scores in ASCII (American Standard Code for Information Interchange) format. This is displayed in Figure 2.2.

FASTQ

```
@HWIEAS269_0001:5:1:1074:5581#NTACAG/1
GTGGGTTATCAGAAC
+HWIEAS269_0001:5:1:1074:5581#NTACAG/1
aaaa`aaaaX^VX]a
```

Figure 2.2. FASTQ file format.

The first line of a FASTA file starts with a '>' symbol and it provides a name and/or a unique identifier of the sequence. The second line is the sequence. The sequences may be protein sequences or nucleotides. An example of a FASTA file format is presented in Figure 2.3.

FASTA

>hsa-mir-155 MI0000681
CUGUUAAUGCUAAUCGUGAUAGGGGUUUUUGCCUCCAACUGACUCCUACAU
AUUAGCAUUAACAG
>hsa-mir-29a MI0000087
AUGACUGAUUUCUUUUGGUGUUCAGAGUCAAUAUAAUUUUCUAGCACCAUC
UGAAAUCGGUUAU

Figure 2.3. FASTA file format.

Detection of miRNAs, isomiRs and other sncRNAs

The first step to detect sncRNAs is to map the reads (in FASTQ or in FASTA format) in a reference genome using an appropriate NGS aligner algorithm. This results in a SAM file, which is a text-based format for storing biological sequences aligned to a reference sequence developed by Heng Li [45] .

The detection of known miRNAs and of isomiRs is usually performed by comparison with the information available at MiRBase [46], a repository that has curated information regarding miRNA and pre-miRNA sequences as well as regarding their respective positioning in the species genome. The detection of other sncRNAs is usually performed by comparison with the RNA central [47], a repository of sequences and genomic coordinates for all families and types of sncRNAs in different species. Most algorithms use comparison by genomic coordinates for sncRNA assignment of mapped reads [7], [18].

Prediction of novel miRNAs

Novel miRNA is performed for mapped reads that are still not assigned to any category of sncRNAs. This task is commonly performed using miRDeep2 software [48].

The hairpin structures of the un-annotated unique reads are analyzed using RNAfold software [49]. Those that formed proper secondary hairpin structures are considered to be the novel miRNAs. The results for the prediction of novel miRNAs are normally presented in the form of a table.

Differential expression

Differential expression analysis consists in taking the normalized read count data and performing statistical analysis to discover quantitative changes in expression levels between experimental groups. Finding genes that are differentially expressed between experimental groups is a vital part of understanding the molecular basis of phenotypic variation. A number of software packages such as edgeR [50], DESeq [51], baySeq [52], and EBSeq [53], have been developed for differential expression analysis of RNA-seq data. The choice of the analysis method has to take into account the experimental design of the experiment. The results for differential expression are normally presented in forms of tables and heatmaps.

2.1.3 Functional analysis

The study of the functional analysis is performed focusing on differentially expressed isomiRs, which implies the prediction of miRNA and isomiRs targets and the probability of the enrichment of biological processes. This allows the user to obtain a comprehensive view of the changes incurred in the cell genetic program.

Files used in functional analysis

FASTA files are common in functional analysis. The format of this file is described in Figure 2.3.

Prediction of miRNA and isomiRs targets

Identification of potential targets may allow unraveling the potential gene network regulated by the expressed miRNAs. To do this, different algorithms have been developed, such as RNAhybrid [54], miRanda [55], TargetScan [56], and Pita [57]. All, except RNAhydrid, have an executable to allow local installation. These algorithms use criteria such as sequence complementarity of the miRNA seed region, thermodynamic stability of mRNA:miRNA duplex, target site conservation among closely related species or other features, to identify the putative miRNA targets. The major use of this prediction tools is the identification of putative target genes for posterior experimental validation. The results concerning target prediction are normally presented in form of a table.

Gene set enrichment analysis

Gene set enrichment analysis is performed upon the list of miRNAs and/or isomiR mRNA targets, and identifies the physiological functions and biological processes that are potentially driven the gene network. The tools normally used to achieve this analysis are the Gene Ontology, Pathway and Protein Domain information (DAVID) [58], and several packages of Bioconductor [59] in R environment [60] such as topGO [61], and GOStats [62]. The pathways identified as significantly enriched are usually represented in diagrams and graphics, whereas gene ontologies are presented in form of a table.

2.2 Existing tools for isomiRs analysis

There are several open source tools for analyzing small-RNA-seq data for identifying isomiRs. Thirteen tools were evaluated, regarding if these included the three main following types of analysis: annotation of sequences, differential expression (DE) and functional impact (see Appendix C for details). The results of the evaluation show that only five of the tools, mirAnalyzer, CPSS, IsomiRex, miR-isomiRExp, and DeAnnIso, have an online server and are able analyze Next-Generation Sequencing (NGS) derived data. However as the miR-isomiRExp [16] website (http://mirisomirexp.aliapp.com/) is not currently working, researchers which have NGS data have only four available tools: mirAnalyzer [7], CPSS [8], IsomiRex [19] and DeAnnIso [18], for which an evaluation of their graphical user interface (GUI) was performed.

Having a GUI is a key property that each tool should possess, because it allows users to interact with the tool through graphical icons and visual indicators, as opposed to text-based interfaces, typed command labels or text navigation. This makes the user's task to learn the functioning of the tool easier and reduces the amount of errors that derive from using multiple tools. GUI takes advantage of the computer's graphics capabilities, enabling (applied to the context of this study) scientists lacking bioinformatics skills to analyze NGS data.

In summary, providing a graphical display is an advantage for a tool, because it is directly related with its usability. Usability is the degree to which a software can be used by specified consumers to achieve quantified objectives [63]. Applied to this project, the tool has a higher usability if its pipeline, for data analysis, is embedded within a web application. If the user runs each step of a pipeline, by himself, directly in the command line, more errors are likely to be made and a bigger amount of time is spent writing the commands with all its arguments (such as folder's names and data files). If there is a defined and automatic chain of events, the tool is much easier to use, and this can be easily achieved with a web application (that provides a GUI), that is organized in focus areas relative to the user's workflow.

A total of 12 analysis features were identified as crucial to be available in a GUI tool for isomiR analysis (see also Table 2.1). Each relevant feature is numbered and accompanied with a small description, being the summary of this comparison.

1. **Creates session ID.**

A session ID is a unique code created at the start of every session that allows easy and fast access to the on-going, or completed, analysis and respective results.

2. **Provides a notification system.**

The user receives a notification when an analysis has started, when it is completed and when an error occurs.

3. **Performs annotation and functional analysis.**

The tool allows the user to perform these two types of analysis (annotation analysis including differential expression), allowing the user to obtain a variety of relevant results.

4. **Annotation analysis and functional analysis can be performed independently.**

The user is given the choice to perform each analysis module independently.

5. **Each analysis is performed for two conditions.**

The user submits two inputs (samples) concerning two experimental conditions.

6. **All parameters necessary to repeat an experience must be known.**

The tool groups the values of the parameters defined in the start of each analysis, so that the user knows this information when starting or repeating an analysis.

7. **Default setting of analysis parameters.**

The values of all parameters are pre-defined, allowing the user to quickly start the analysis.

8. **Example files are available to run each analysis.**

Example files are provided for each analysis (annotation and functional), allowing the user to assess the potential of tool performing quick start analysis.

9. **Display of a progress bar that changes accordingly to evolution of the analysis.**

The workflow allows the user to understand, after the start of an analysis, which step of it is being performed.

10. **Clear and concise help about the annotation and functional analyses.**

The tool must provide help and documentation to the user. This information must easy to search, be focused on the user's current task, list concrete steps to be carried out, and not be too long.

11. **Results can be downloaded independently or in a bundle.**

The results can be downloaded independently, or they can be downloaded in a ZIP file. This ZIP file also has all the elements necessary for repeating the experience.

12. **Provides a results index.**

The results' index groups the results present in each webpage, allowing an easy access to any result in the index.

After the comparison between the four tools, displayed in Table 2.1,it is possible to observe that both isomiRex and mirAnalyzer, in terms of functionalities, allow performing an incomplete set of analysis, not even performing annotation of sncRNAs and functional analysis. CPSS does not allow the user to analyze data from different experimental conditions, which is usually applied to any experimental design.

Table 2.1. Comparison of functionalities between the isomiR analysis tools.

Requirements	isomiRex	mirAnalyzer	CPSS	DeAnnIso
1. Creates session ID			**x**	**x**
2. Provides a notification system			**x**	**x**
3. Performs annotation and functional analysis			**x**	**x**
4. Annotation analysis and functional analysis can be performed independently				**x**
5. Each analysis is performed for two conditions	**x**		**x**	**x**
6. All the parameters necessary to the repeat of the analysis must known		**x**	**x**	**x**
7. Default setting of analysis parameters		**x**	**x**	**x**
8. Example files are available to run each analysis			**x**	**x**
9. Display of a progress bar that changes accordingly to evolution of the analysis			**x**	**x**
10. Clear and concise help about the annotation and functional analyses			**x**	
11. Results can be downloaded independently or in a bundle				**x**
12. Provides a results index			**x**	**x**

DeAnnIso is the tool that offers more functionalities. For this reason, it was attempted to measure the time taken by DeAnnIso to perform a single analysis, the equivalent to the annotation analysis module. This, however, could not be achieved, since the file format is different (FASTA format instead of FASTQ format used in IsomiR Window), the number of files for experimental condition is limited to one, and the size of the submitted files cannot exceed 50 MBytes, which is not compatible with the size of the data of an NGS library.

2.3 Web development

When discussing web development, it is important to highlight the two parts that, together, form a web application. In one part is the client, represented by the frontend that utilizes Uniform Resource Identifiers (URIs) to access information contained in the web server, located in the backend, though, in most cases, the Hyper Transfer Protocol (HTTP). The frontend supports the interaction with the user and it is built through the use of three different languages: HTML, which defines the document structure, CSS that establishes rules for the layout of the different document elements and JavaScript, which controls the behavior of those elements [64].

The second part is the backend, where the server and all its connections are based. The server's function is to accept a request from the client and reply to this request, usually by presenting a requested web page [64]. Instead of replying with a web page, the server can pass information to the frontend though the use of web services. These are called by AJAX technology, in the frontend. Additionally, the

server has common gateway interface that can use several languages, such as PHP. The server is also responsible of establishing the connection with the database whenever there is a need to insert, update or delete data.

There are many building blocks that can be used in web development, which are introduced in the section below. In addition, the concept of frameworks, the base of the web application, is explained, as well as which frameworks are most commonly used nowadays.

2.3.1 Client building blocks

In this section, the building blocks used to produce the frontend are presented.

HTML

HTML5 [65] is a markup language used for defining the structure of web pages, making their contents interpretable by the browser. The new version, 5, introduces new changes that assist in the development of web applications. One of the most addressed changes in HTML5 is the functionality that allows browser native support for Scalable Vector Graphics (SVG) graphics.
Elements such as <table> and <form> are often used to enrich the semantic content of the documents. They allow the creation of tables and forms, respectively, within the document. Additionally, these two elements can interact, allowing the user to insert a table within a form to store any relevant information that this table might contain. Other elements such as <a> are also very useful, allowing the user to navigate to a different webpage or to download files available in the webpage.

CSS

CSS [66] is a language that describes the presentation of HTML documents. These rules can be applied to a single HTML element (with the id attribute), applied to a group of elements (with the class attribute) or to elements that share the same tag name.

The CSS should be in a separate file so that the rules defined can be applied to all the documents that compose the application, making the all pages visually coherent. Additionally, because the CSS file has a small size, it is possible to quickly load the defined rules and store the file in the browser's cache, so it can be reused for all HTML pages that share that same CSS file, without contacting the web server.

JavaScript

Through the years, HTML developed the ability to support event handlers, which are usually small functions that are executed when a specific event occurs in the browser. Examples of these events are actions such as clicking on a button, stop pressing a key when typing in a field, or even the reloading of the webpage. The implementation of these events handlers requires a programming language such as JavaScript [67], that is capable of being executed in the client side of a web application. Therefore, JavaScript can read and change the contents of HTML elements, as well as the style of these elements, through the modification of CSS rules. JavaScript is also useful for data validation, controlling the user's input data before this data is passed to the server.

AJAX

Web applications characterized by responsive user interfaces and by interactive features represent an easier and more functional mode for the development of these applications, improving the user's experience. AJAX technology [68] stands for Asynchronous JavaScript and XML, although Extensible Markup Language (XML [69]) isn't necessarily used. Using AJAX, an HTML page can perform asynchronous calls, through HTTP, to the server and loads the content of the response, which can be XML, HTML, simple text or JavaScript objects. This response can be used to alter the displayed document's content without being necessary to reload the webpage.

An example for the use of AJAX is a progress bar, which is being updated as it receives information from the server, through different calls. Another example of AJAX is the auto-completion of words in a field that is being filled by the user.

2.3.2 Server building blocks

In this section, the building blocks used to produce the backend are presented.

Web Services

Web services are a group of methods that are accessed and invoked by other programs using web technologies. They are mainly used in system integration and in the communication between applications, allowing new systems to interact with the existent ones and also allowing compatibility of the developed systems in different platforms [70]. They are also used to transmit the data between the database and the frontend in a bi-directional path.

There are several ways to implement these services, however two can be highlighted: SOAP (Simple Object Access Protocol) and REST (Representational State Transfer) [71].

To access information of a web server it is necessary to know which type of desired request and information. There are several HTTP methods that can be used to perform requests, and this is achieved by URIs (Uniform Resource Identifier) [72]. Four HTTP requests are the most relevant and they are:

- GET – obtain information of a resource;
- POST – adds information to a resource;
- PUT – updates information of a resource;
- DELETE – deletes a resource.

PHP

PHP [73] is a server scripting language, and a powerful tool for developing dynamic and interactive web pages. It is widely-used, free, and supports different types of operating systems, being one of the first languages to be directly integrated with HTML. The PHP code is interpreted by the server, which generates a dynamic output - output that varies according to the interaction with the users and the data that is being manipulated.

PHP supports user sessions, database access, and is capable of invoking other programs that are installed in the server.

2.3.3 Model-view-controller frameworks

For the web application, desired for this project, there are several frameworks that can be used in its development. In this section is described the importance of using a framework, a brief introduction to the Model-View-Controller (MVC) architecture pattern, what elements to consider when choosing a framework, and a summary of the most popular frameworks used nowadays, such as Laravel [74], Symfony [75] and CodeIgniter [76].

Usage of a framework

PHP is a server scripting language that provides services for handling email, database interaction, and more. Because of its power and flexibility, coding in plain PHP to build a web application works, but there are several activities that can go wrong, such as developing meandering code that performs many functions in one place. This leads to disorganized code, which is a big maintenance problem, because the code it is not as easy to reuse and understand. Testing is another problem that arises when using plain PHP code. Because the code is not necessarily divided into fine units, it can be extremely

difficult to test this code. Additionally, when working in a team, in a regular PHP application, it can be very difficult to have multiple people operating on different aspects of the application at the same time.

Frameworks offer solutions for many of these problems. The main advantage of using a framework is that it provides a guideline to be followed in the development of the application. This guideline mainly consists in dividing the code into several files that follow a documented organization, making it easy to reuse them throughout in the different parts of the application. This division also makes the task of changing a piece of the code much easier, which is a plus for updates of the application in the future. Another key advantage of using a framework is that it allows multiple people on a team to work on the application at the same time.

One thing to consider when using a framework is that it comes with a learning curve. Whereas a single PHP page can contain all the domain logic, database access, and the data presentation, in a framework, all the parts have to be separated into different components, which can take additional time. The developer has to learn how the code must be organized and what is offered by the framework, namely its Application Programming Interface (API). Despite the learning curve, frameworks lead to better organization of the code, making the development and maintenance of the application more productive.

Understanding MVC: Model-View-Controller

At the core of every framework is the concept of patterns or architectural patterns. MVC stands for Model-View-Controller, and this pattern represents the concept of dividing an application into three coherent aspects: the Model (Data layer), the View (User Interface layer), and the Controller (mediates user interface interactions and updates to the Model) [77]. In Figure 2.4 is shown a diagram representing the MVC pattern.

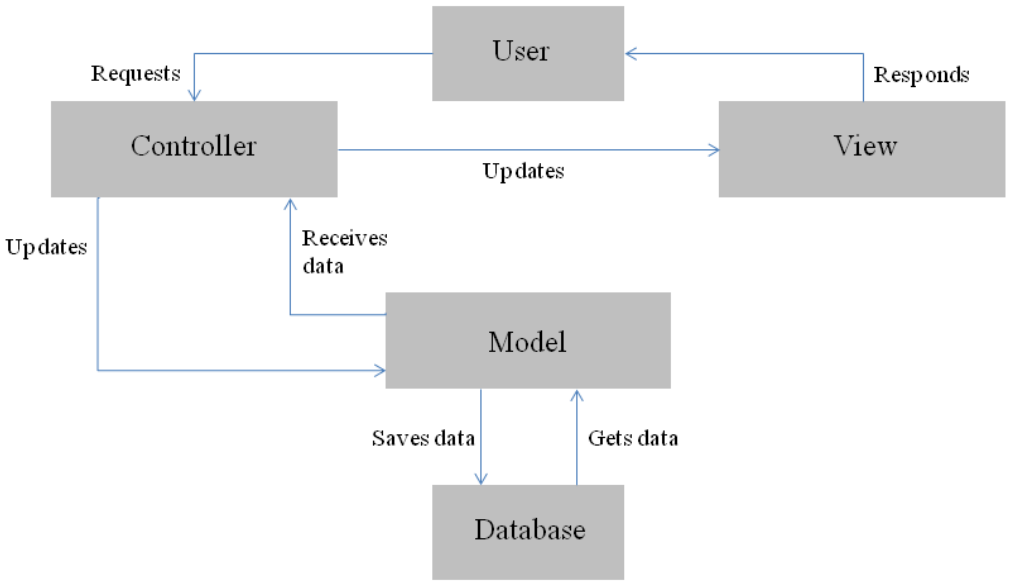


Figure 2.4. MVC architectural pattern [78].

Each framework implements the pattern in a specific way, defining which changes affect the different aspects and coding practices are embedded in the framework.

Framework programmatic concepts

There are some core development concepts that need to be understood for the evaluation of which framework to choose and for the learning of how to use the chosen framework.

The first concept is Pattern usage: whether the framework strictly enforces the usage of MVC and other patterns, or if it is more flexible in terms of how the code is organized and naming conventions. It is also important to analyze the classes that are included within a framework, and these mark the difference when developing an application. They provide shortcuts to building several functionalities, that otherwise would have to be built from scratch. Therefore, analyzing which classes are available in a specific framework and, whether the users are obliged to use them is a key part of the process.

Another concept to keep in mind is Modularity. It is important to assess if the framework accepts modules from other frameworks, allowing the developer to use these separate modules for the growth and better functioning of the application.

The final concept is Standards. The PHP Framework Interoperability Group (PHP-FIG) [79], a consortium of PHP developers, created in 2009, that come from all of the major MVC frameworks, has created the PHP Standards Recommendations (PSRs). An important standard is the Composer. Composer is a tool for managing dependencies in PHP applications, downloading them automatically. Frameworks that support it, make it much easier to use their elements, as well as install them in the development environment.

Understanding how a framework incorporates pattern usage, code organization, data handling mechanisms, availability of components, modularity, user interface helpers, and standards, helps in choosing a framework and using it.

Comparing concepts

In addition to the programmatic concepts, it is also important to evaluate other key elements when deciding which framework to use. They include licensing, whether the framework is light weight or heavy weight, documentation and community support, complexity and performance.

The type of license that the framework uses can be an important decision factor. Some make it easier to be bundled in software packages, others do not.

As for the weight, some frameworks are light weight and simply provide a basic structure for the developer to follow, whereas others provide many tools, making the application very tightly tied to that framework, and will also restricted by the framework's limitations for performance and features. In fact, the developer is not obliged to use these tools, but the community for this type of frameworks tends to use them, not only for the existent support but also because of the official documentation.

It is important to assess the documentation of the framework, verify if there are complementary books available, and if there are any online courses. This concept is relevant because, once a framework is chosen, it will determine how easy it will be for the developer to learn it.

Additionally, the community and how active developers are in that community should also be evaluated. A common quantifier of this element is how many Stack Overflow [80] questions exist, and are answered, for the framework of interest.

Scalability and performance are also elements to have into consideration. The weight of these considerations, however, depends on the type of application that is being built, and how many users it will have.

Relevant frameworks

Symfony framework

Symfony [75] has long been one of the most popular PHP frameworks for several reasons. It has a solid code base community and good documentation, it is flexible, has a toolbox for rapid application development features, and its learning curve is relatively slow. Symfony is backed by the company SensioLabs, is currently on version 3.2 and it has been distributed with the MIT license.

Previous versions of Symfony have leaned towards the full-stacked model. Currently it is possible to use the full-stack framework or just some of its modules separately, making it flexible. These modules

can be installed with the Composer PHP dependency manager, and be used in other frameworks such as Laravel, which is another is described later in this section. In addition, Symfony helps programmers create scalable applications that are flexible to changing business requirements. Symfony's ability to use different Object-relational Mapping (ORM) systems, is one of the reasons why this framework is one of the top choices for programmers.

Even though Symfony has all these features, beginner level developers might find it complicated due to its complex structure.

In summary, Symfony excels because its modules are portable and the full-stack provides a lot of power as well as flexibility. However, it can be challenging to understand and learn at first.

CodeIgniter framework

CodeIgniter (CI) [76] is commonly used by PHP developers who prefer a lighter weight framework. In 2014, CI management changed to the British Columbia Institute of Technology. CI had a couple of different licenses, but starting in the 3.0 version, an MIT license was adopted.

This framework has a very small footprint and it is not strictly based on the MVC development pattern, but provides many components that help with rapid application development. Because of its light weight, it can be very flexible, which can help when working with hosted environments. It is also highly decoupled, which means a developer can use a much or as little of the framework as required. It is not strict in terms of enforcing usage or coding rules.

CI was a very popular framework at one point, but it stalled in development during a large period, and for that reason some aspects of the framework are somewhat behind some of the other frameworks. It was not until version number 3.1.5, released in June of 2017 that CI integrated Composer. Additionally, CI is not a member of the PHP-FIG and it doesn't use built-in namespaces.

In summary, CI is easier to begin using by someone unfamiliar with frameworks, supports object-oriented programming and structuring large applications. Its installation is easy and the framework itself is fast (because of its small size, when compared to modern giants like Symfony and Laravel). However, development stagnated for a long time. Lots of modern PHP practices were adopted during this time, leaving CodeIgniter behind. It lacks modern namespace and autoloader use.

Laravel framework

Laravel [74] is recent to the PHP Framework space, having been released in 2011. It has already moved to version 5 and acquired an active and productive user base. This framework is one of the top used PHP frameworks and one of the reasons developers are finding Laravel so compelling is that it leverages the latest generation of programming libraries. The version of Laravel, when developing the current project, uses a number of different components from Symfony. Interestingly, the flow goes both ways, being possible for Symfony users to work with Laravel components. The reason for this efficient development is the fact that Laravel supports Composer.

Database access and management are an important aspect of most web applications. Laravel's database component is called Eloquent ORM. Eloquent was created by Laravel, but as it is a self-contained package, it can be used outside of the framework. Eloquent works with an ActiveRecord pattern, which allows easy inserting, updating and deleting of records.

Another benefit of using Laravel is its command line interface called Artisan. The framework is REST-friendly and uses many JavaScript constructs, which is why it is one of the best frameworks for developing AJAX applications. Laravel ships with its own template engine called Blade and uses a very intuitive routing mechanism. It also has an official website, Laracasts, that offers many video tutorials.

A disadvantage of Laravel is that its core files are stored within its namespace and not all of these files use a namespace slash (a \) for calling another core file. This makes it difficult for the developers to extend some classes. This is, however, not a big issue for all the developers.

Laravel really has taken a good look around and incorporated the best features from existing PHP libraries and from those of other platforms like Ruby on Rails, as well as developing its own systems internally.

Comparing Symfony, CodeIgniter and Laravel

This section displays a table (Table 2.2) that contains a comparison of the three frameworks mentioned above, based on the features described.

Table 2.2. Comparison between MVC frameworks.

Features	Symfony	CodeIgniter	Laravel
Minimum PHP	5.5.9	5.2.4	5.5.9
Support and training	Official documentation, lynda	Official documentation, lynda	Official documentation, lynda, Laracasts
License	MIT	MIT	MIT
Performance	Medium	High	Medium
Project activity	High	Medium	High
Community size	35K	15K	40K
Developer's framework experience	Medium	Low	Low
Year of creation	2005	2006	2011
Latest version	3.3.2	3.1.3	5.4
Integrated ORM	Doctrine, Propel	Doctorine, DataMapper, Gas	Eloquent

After the comparison between the three frameworks, Laravel was the one chosen for the web application of this project. This framework is easier to learn, uses composer to manage its dependencies, is REST-friendly and has a good community support.

2.4 Summary

In this chapter, a brief description of miRNAs and their isoforms was provided, and why it is so important to study these sncRNAs. Additionally, the main steps for the bioinformatics analysis of this data were also presented. The tools that perform isomiR analysis were described. As the final section of this chapter, the building blocks and frameworks used in web development were also described.

In the following chapter, the description of the development of the IsomiR Window system is provided, where several topics will be addressed, such as system architecture, application structure, as well as the created web services and the display of web interface.

Chapter 3 – Web application for the IsomiR Window tool

This project consisted in the development of a frontend and backend of a web application, that allows the user to perform one of the two modules of analysis, annotation (coupled with differential expression) and functional, independently or perform functional analysis based in the isomiRs found in the annotation analysis.

One of the concerns in the development of the web application was that any user, with any type of background, would be able to easily manipulate it, without losing the context of the analysis. To achieve this goal, several help sections were created and whenever an error occurs, the user receives messages documenting the cause and advice regarding problem solution. The chosen name for the complete system, web application and pipeline, is IsomiR Window tool.

The next sections in this chapter describe the system architecture, application structure and design, backend, which includes a description of the created web services, and finally the frontend that demonstrates the main functionalities available in the user interface.

3.1 System architecture

It is important to describe the components involved in the system, and of their interactions, before explaining its implementation. The system architecture followed a client-server approach. The server consists in an archetypal model of web service stacks called LAMP [81], that stands for the Linux operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language.

The system was structured to allow a connection between the different layers, where the client (browser) makes requests to the server, including asynchronously requests via AJAX. These are then processed in backend by different tools and technologies. Requests that concern data analysis are processed by a Perl-base pipeline layer (not in the scope of this thesis). The outcome of this processing is the return of the obtained outputs to the user.

Figure 3.1 displays the system architecture and it is divided into three main layers: Frontend, Backend and Pipeline (Perl-based). The frontend is responsible to display an interactive interface to the user by the browser. This is possible through several technologies, such as HTML5, AJAX, JS, CSS and a JS library Highcharts.

The domain logic in the backend receives the information by the frontend and performs several actions on this information. The web services are within the domain logic, because they invoke some of its functions. The domain logic is responsible for storing data in the database and for invoking several Perl scripts in the pipeline layer, which perform (user inserted) data processing. Each script, consisting in custom developed algorithms and invoking third-party software packages, is executed to accomplish each phase in the treatment of the data.

The chosen framework for the development of this project was Laravel [74], which is responsible for the domain logic code organization and functionalities. After the comparison between the most relevant MCV frameworks in Section 2.3.3, Laravel was the most suitable candidate.

This framework was selected since it allows a stepping-stone learning process for beginner users, providing a simpler syntax and good learning materials. PhpMyAdmin is a web application that is accessible through the browser for a graphical visualization and administration of the database.

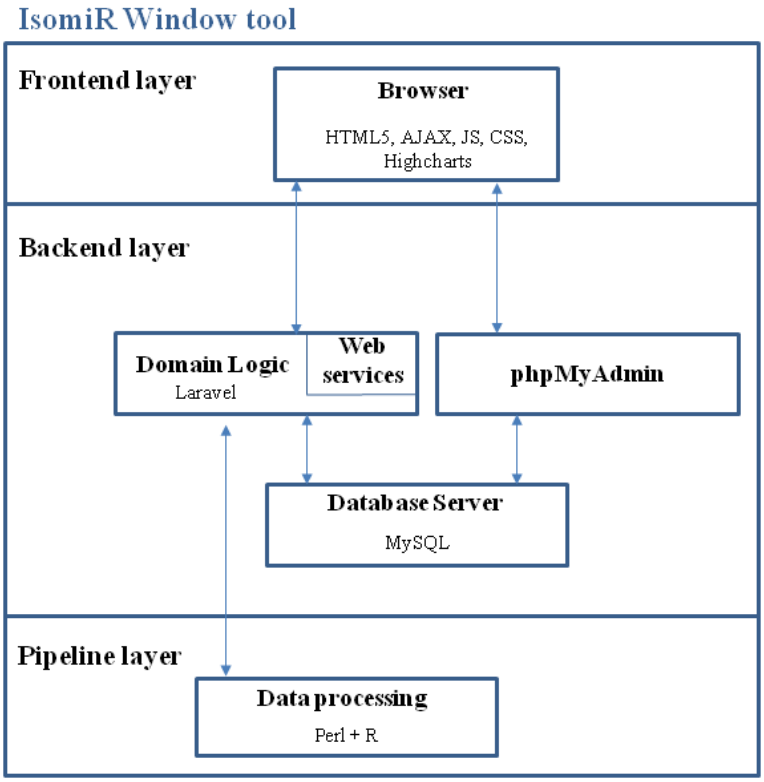


Figure 3.1. Architecture of the IsomiR Window tool.

3.2 Application design and structure

Establishing the design of an application is an important step in its development. It is good to have a starting point, allowing the developer to know which component of the application to develop next as described below in Figure 3.2.

Additionally, after the application is completed it is important to understand where it main components are. It is important to see how the aspects of the MVC pattern are organized, along with organization of the output files of the pipeline layer.

3.2.1 Application design

To design the application workflow, a User Environment Design (UED) was developed, which is shown in Figure 3.2. The UED defines how the application should behave and organizes its functions and work objects in a way that makes sense for the user [21]. It keeps the user's work coherent by defining several focus areas to support a particular type of work.

The UED is helpful because when visual representations do not yet exist, it becomes harder to look across the entire system, and decide if its parts are coherent and where a new function should go [21].

Figure 3.2 shows the UED for the IsomiR Window web application. The first focus area (Home) is where the user can start a new analysis or to see the results of a prior analysis. In the first case, the user can choose to proceed to focus area number 2 (annotation analysis and DE configuration). Similarly, the user can proceed to focus area number 5 (functional analysis configuration).

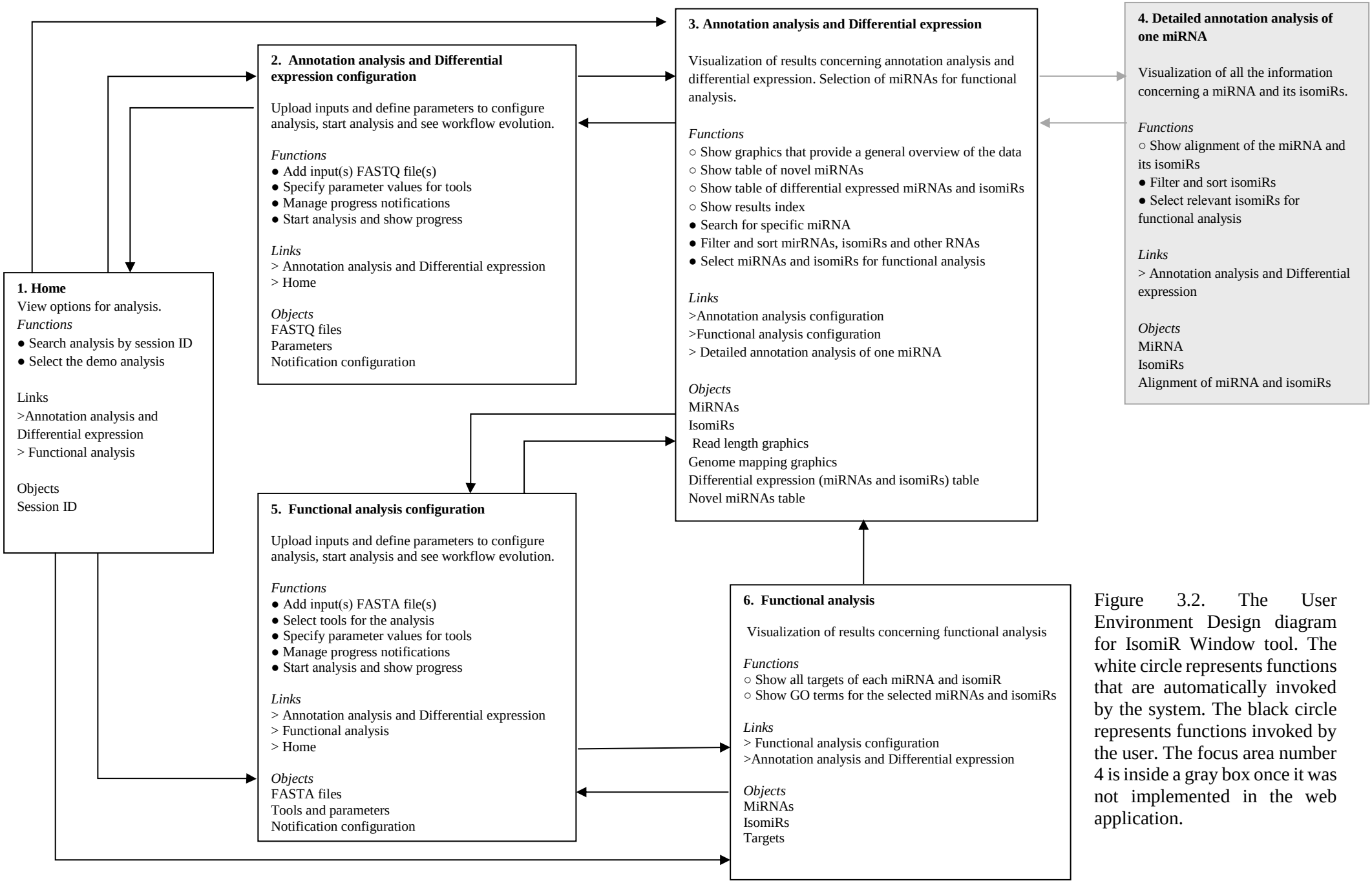


Figure 3.2. The User Environment Design diagram for IsomiR Window tool. The white circle represents functions that are automatically invoked by the system. The black circle represents functions invoked by the user. The focus area number 4 is inside a gray box once it was not implemented in the web application.

In the second scenario, in which the user wants to consult a prior analysis, it is possible to proceed to focus areas number 3 and 6.

The most natural path for a user to follow was considered to be from focus areas 1 to 6, step by step. When the user reaches focus area number 3, the annotation analysis and DE is completed it is possible to end the analysis at this stage or to proceed to focus area number 5, corresponding to the functional analysis, using as input the isomiRs found and selected in the previous area. After the functional analysis is completed, in focus area number 6 it is possible to return to focus area number 3 to change the previous selection, allowing the functional analysis of different isomiRs. There are several paths the user can chose to go through, however, the ones described above were considered the most relevant.

Focus area number 4, despite being present in Figure 3.2, was not developed in the web application. This was because the outputs produced by the scripts in the pipeline layer did not provide the desired level of detail. Therefore, because specific information about each miRNA (and isomiR) was not produced, there was no content available for the development of the functionalities listed in focus area 4. However, two of its functionalities, Filter and sort isomiRs, and Select relevant isomiRs for functional analysis were implemented in focus area number 3.

3.2.2 Application structure

The Laravel framework is divided into ten main folders, however, only the ones considered to be the most relevant to this project will be described here. Additionally, inside these folders, there are certain files that deserve special attention and explanation.

The first file to mention is the environment file, called `.env`, which is where the configuration variables are attributed values based on the environment where the application is running. In this file, for example, is where the type of database connection is established, as well as the database host address and name.

The `routes` folder contains several files, but it is of particular interest the one named `web.php`, where all the routes of the application are defined. The route's purpose is to match certain Uniform Resource Locators (URLs) that invoke a function within a controller.

In the folder `database`, one of its subfolders, `migrations`, contains the PHP files that, through the Artisan console (the command-line interface included in Laravel), allow the creation of the tables in the database.

Another folder is `app`**,** where all the files corresponding to controllers and models are stored. The models created directly related to each table created in the database, though migrations. They are called `AnnotationForm.php`, `FunctionalForm.php` and `Session.php`. The controllers are stored inside the `app/Http/Controllers` and they can be divided into 3 groups, annotation analysis, functional analysis, and general pages, which is composed by one controller called `PagesController.php`. For the annotation analysis module there are three controllers. The first controller, `AnnotationForms Controller.php`, stores input data provided by the user, whether it is uploaded by browser or stored manually. Additionally, this controller is responsible for saving the information relative to the analysis settings in the database. The `AnnotationProgressController.php` is responsible for executing the Perl scripts in the pipeline layer, to initiate input data processing. The last controller, `Annotation ResultsController.php`, processes the data files that were created as outputs of the Perl scripts and transforms them into information that can be received and understood by the browser for a graphical display. The same file structure and responsibilities apply for the functional analysis controllers group, replacing the name "Annotation" for "Functional".

In the folder `resources` is where the views are stored as well as the pipeline Perl scripts. Similarly to the `Controllers` folder, the `views` folder has 3 major groups, containing about 5 different views each. Two groups concern annotation and functional analysis, being the Views very identical, varying

only in content. The third group is relative to the layouts present in all the other views. This is possible because Blade, the template engine provided in Laravel, allows template inheritance and sections.

Still in the `resources` folder, is a subfolder called `assets`, which stores the data processing scripts visible in Figure 3.3. The script `find_ncRNAs.pl` is responsible for the alignment of reads and detection of sncRNAs. The `mirdeep.pl` script performs the prediction of novel miRNAs, and the `find_isomiRs.pl` detects miRNAs and isomiRs. The script called `charts.pl` does not process input data itself, but processes outputs produced by the other scripts, of annotation analysis, to create data files that contain the information used to build the graphics displayed in the web application. The scripts `deseq.pl` and `Rdeseq.R` are responsible for differential expression (DE) assessment, which are incorporated in the annotation analysis module. The functional analysis module has two phases, target prediction and gene enrichment, both accomplished by the `functional.pl` script. The `RtopGo.R` script is called within `functional.pl`. As mentioned earlier, the development of these scripts was performed within the scope of IsomiR Window project but outside the scope of this thesis. Therefore, for more detail about the scripts consult the thesis by Inês Viegas [82].

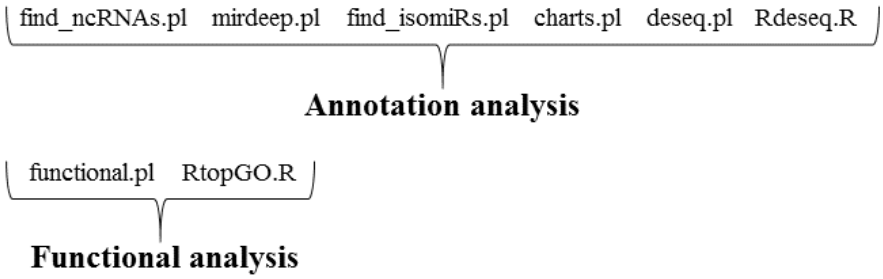


Figure 3.3. Pipeline scripts for annotation and functional analysis.

Another folder, `storage/app`, stores all the data files relative to each session, including both possible analyses. Each sub-folder of `app` is called after the session ID. Simulating that the user performs annotation analysis (with session ID equal to 1), including the prediction of novel miRNAs (optional), followed by functional analysis, what happens is that, inside the session folder, in this case folder `1`, there are three sub-folders, two named after each analysis (annotation and functional analysis) and another called `Mirdeep`. Inside those, there are several others, containing inputs, results and data processing messages. The organization of the folder for the complete session is presented in Figure 3.4 and Figure 3.5.

In Figure 3.4, it is possible to observe the three main folders: `AnnotationAnalysis`, `FunctionalAnalysis` and `Mirdeep`. For the example displayed in Figure 3.4 andFigure 3.5, two input files were uploaded for experimental condition 1 and one input file for experimental condition 2. The `AnnotationAnalysis` folder is the most extent of the three, which is why it is fully represented in Figure 3.5, being this figure discussed later.

In the `FunctionalAnalysis` folder it is possible to visualize all its sub-folders and files. Two files in this folder are the `isomiRsFile.fasta` and `TableTopGo.txt`. The file `isomiRsFile.fasta` is used as input for the `functional.pl` script, which is the one (combined with the `RtopGO.R` script) that performs the functional analysis. The other file in this directory is `TableTopGO.txt`, which is one of the outputs of this analysis, and the one that is displayed in the web application.

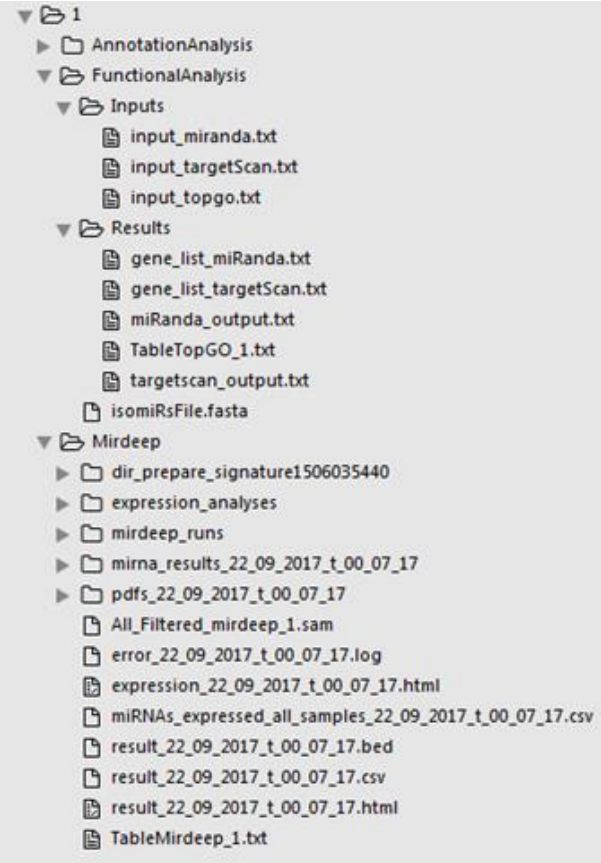


Figure 3.4. General folder organization for Session 1.

Prediction of novel miRNAs is an analysis, performed within annotation analysis, which usually requires a longer period (≈3hours, estimation obtained in a Linux machine with 16 GBytes of RAM and 4 processor cores). Although this analysis contributes to increase the time of the pipeline processing, it is highly relevant, especially in the context of genomes for which the identification of the miRNA catalogue is still in its beginning. For this reason, when configuring the settings for annotation analysis, the user may choose to perform the prediction of novel miRNAs, which is computed using the MirDeep2 third-party software package [48]. This can be done automatically (the web application is responsible for its implementation) or manually (the script and instructions is available for the user to download), allowing the user to run this analysis separately.

The manual approach was developed for the case in which the user is not interested in predicting novel miRNAs for the time of the current analysis, allowing the user to perform this analysis on a different occasion, speeding up the generation of results for the remaining analysis performed by the IsomiR Window tool. For this reason, the `Mirdeep` folder is located outside of the annotation analysis folder, for easier access to the input file, named `All_ Filtered_mirdeep_1.sam`, that is to be used in the script made available to the user. If the user chooses to perform the prediction of novel miRNAs automatically, all the outputs of the analysis (created by the script `mirdeep.pl`) are stored inside the `Mirdeep` folder, and the one called `TableMirdeep_1.txt` is displayed in the web application.

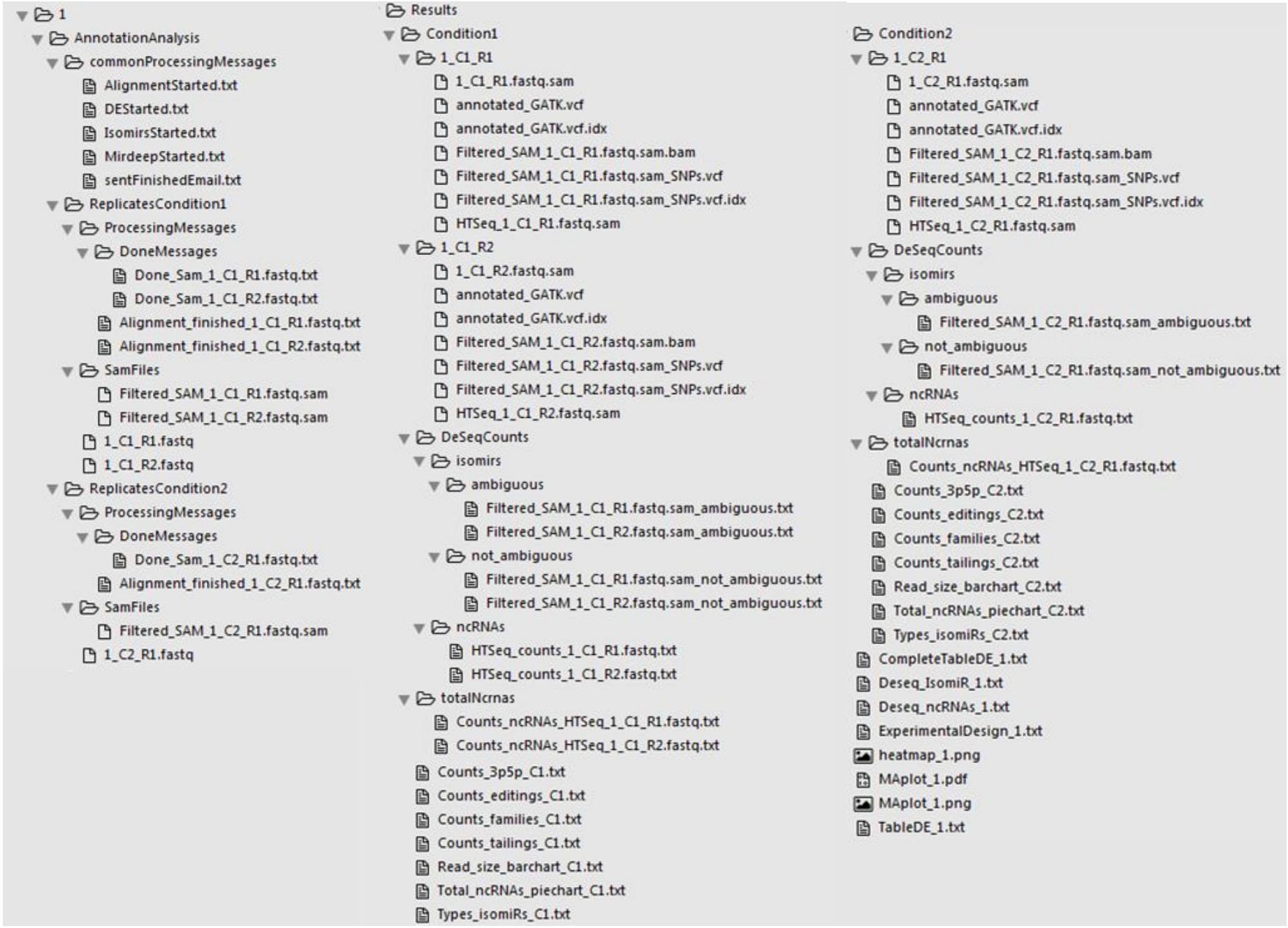


Figure 3.5. Files and folders for session 1, which has two conditions, with two files inserted for condition 1 and one file for condition 2.

Now focusing on the annotation analysis and its folder organization, represented in Figure 3.5, the two sub-folders `ReplicatesCondition1` and `ReplicatesCondition2` inside `/1/AnnotationAnalysis`, are where the user's uploaded files are stored for each experimental condition, one and two respectively. These files are used as inputs for the script `find_ncRNAs.pl` and, as a result, several other files are produced. In particular, the files in the `SamFiles` folder (in both the `ReplicatesCondition1` and `ReplicatesCondition2` folders) are used as inputs for the scripts `mirdeep.pl` and `find_isomiRs.pl`.

Another folder is `Results`, which is divided into two sub-folders `Condition1` and `Condition2`, where the results produced by the scripts concerning the replicates of each condition, one and two respectively, are stored. Inside `Condition1` and `Condition2`, sub-folders of `Results`, store several output files, such as `Counts_3p5p_C1.txt` and `Counts_3p5p_C2.txt`, which store information regarding sequencing counts of isomiRs derived from the pre-miRNA hairpin arm 3p or from arm 5p and, of all the replicates within the experimental conditions. The files stored in the `Results` folder are used as input for the production of the pie and bar charts that are displayed in the web application frontend. Still inside `Results`, but outside of both `Condition1` and `Condition2` sub-folders, are the results from both experimental conditions, such as the `TableDE_1.txt` file. The files inside the directory `/1/AnnotationAnalysis/Results/Condition1{orCondition2}/DeseqCounts/isomirs/not_ambiguous` are used as inputs for the deseq.pl pipeline Perl script.

The only files in Figure 3.4 and Figure 3.5 that are inserted by the user are the ones called `1_C1_R1.fastq`, `1_C1_R2.fastq` and `1_C2_R1.fastq`, these for annotation analysis, and `isomiRsFile.fasta` for functional analysis. However, the user only inserts the latter if functional analysis is done independently of the annotation analysis. Otherwise, this file is created internally when the user chooses to proceed to functional analysis based on the isomiRs found in the annotation analysis.

Also inside the application's framework is the `public` folder, which contains the files for the presentation of the web pages, such as CSS files, images and JavaScript files.

3.3 Backend

The backend of the web application consists in a database designed to store the information about session initiated by the user, along with settings of the analyses performed during that session.

3.3.1 Database

It is necessary to have a storage system for the data collected by the web application. This system must be prepared for constant consulting and update to maintain data integrity. Therefore, a relational database was created, using MySQL, and it stores part of the data processed by the web application. The name of the database is `isomirWindowdb` and it has three tables (entities), one for annotation analysis, another for functional analysis, and a last one that bridges both types of submissions. The table `Sessions`, mainly stores the session ID that will be used in each analysis. The figure below displays the associations between the entities and their corresponding attributes.

From Figure 3.6, it is possible to verify that an entry in the `Sessions` table creates another entry in one of the other two or even in both. The most likely and logical scenario that can arise is the user starting in the annotation analysis. When this happens, 2 entries are created, the first one in the `Sessions` table and the second in the `Annotation Forms` table. If the user chooses to proceed for functional analysis, only one entry is created and it is in the `Functional Forms` table.

In the table `Sessions` there is the `id` attribute, which is automatically attributed by MySQL, `species` (name referent to the species), `email` (e-mail address) and the attributes `created at` and

`updated at`, which store the hour and date of the creation and modification, respectively, of each table entry. These two attributes are also automatically inserted by MySQL and are present in the three database tables.

In the table `Annotation Forms` there is also the `id` attribute, inherited from `Sessions` table, and the attributes that were defined as analysis settings. The first is `mismatches` (number of mismatches acceptable), `genomicHits` (number of genomics hits acceptable), `experimentalDesign` (whether the provided samples are paired or unpaired) and `significance` (p-value to be used in differential expression). The attribute `predictMirnas` is used to save the user's decision about performing the prediction of novel miRNAs within the annotation analysis. The attributes `numberFilesC1` and `numberFilesC2` are used to store the number of files inserted by the user for each experimental condition. There is also the attribute `notes`, which stores a message related to the annotation analysis.

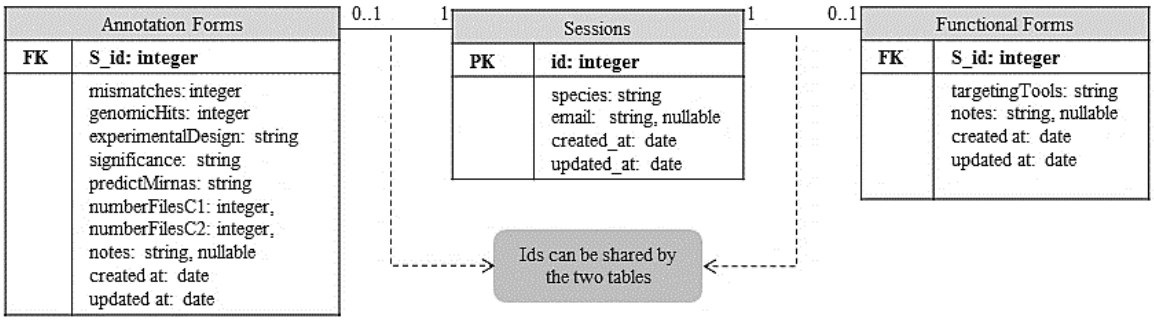


Figure 3.6. Relational database structure.

In the `Functional Forms` table there is the `id` attribute, also inherited from `Sessions` table and the attributes `targetingTools` (names of the targeting tools to be used in the analysis) and `notes`, which stores a message related to the functional analysis.

All the attributes of the three tables, except for the `id`s and timestamps (`created at` and `updated at`) are filled with data that the user inserts in the forms present in the views `create` from annotation and functional analysis.

3.3.2 Web services

REST (Representational State Transfer) protocol was chosen to implement the web services in this project. One of the reasons for this is the fact that the framework being used to build the web application, Laravel, is REST-friendly, as mentioned in the previous chapter. Additionally, REST is relatively easy to implement and maintain, it can return data in multiple formats, such as HTML, JSON, XML, Plain text, PDF, and more, and can be consumed by any client, even a web browser with AJAX and JavaScript. SOAP, on the other hand, is harder to implement for web and mobile developers, and only supports the XML format.

In the current project, most web services are supported by GET requests, except for the upload of data files from the web interface, which is carried out via a POST request. DELETE and PUT requests are not used in this project, since there is no intention or permission to change or delete data. In the cases when the user uploads incorrect data, the solution is to create a new user and upload new data. In the list below are displayed the URIs for the web services of the application.

1. **/annotation/session/create**
   - GET Method;
   - Creates an entry in the database tables `Sessions` and `AnnotationForms`. Returns the id to be used as session ID.

2. **/annotation/session/{sessionID}/check/EndOfAlignment**
   - GET Method;
   - Parameters:
     - sessionID – ID of the current analysis;
   - Verifies if output files concerning the first step (alignment) in the pipeline are created and reads them. This information is returned to the browser for updating the progress bar.

3. **/annotation/session/{sessionID}/check/EndOfDE**
   - GET Method;
   - Parameters:
     - sessionID – ID of the current analysis;
   - Verifies if output files concerning differential expression in the pipeline are created and reads them. This information is returned to the browser for updating the progress bar.

4. **/annotation/session/{sessionID}/check/EndOfIsomiRs**
   - GET Method;
   - Parameters:
     - sessionID – ID of the current analysis;
   - Verifies if output files concerning the detection of isomiRs in the pipeline created and reads them. This information is returned to the browser for updating the progress bar.

5. **/annotation/session/{sessionID}/check/EndOfMirdeep**
   - GET Method;
   - Parameters:
     - sessionID – ID of the current analysis;
   - Verifies if output files concerning the prediction of novel miRNAs in the pipeline are created and reads them. This information is returned to the browser for updating the progress bar.

6. **/annotation/session/{sessionID}/check/EndOfNcRNA**
   - GET Method;
   - Parameters:
     - sessionID – ID of the current analysis;
   - Verifies if output files concerning the second step (detection of ncRNAs) in the pipeline are created and reads them. This information is returned to the browser for updating the progress bar.

7. **/annotation/session/{sessionID}/getBarChart/{filename}/{condition}**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
        - filename – name of the output file to be read;
        - condition – declares if the pie chart if for condition 1 or 2.
    - Gets all the lines of the output file, in form of an array, for it to be built in form of a bar chart in the browser by JavaScript and Highcharts library.

8. **/annotation/session/{sessionID}/getBarChart2/{filename}/{condition}**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
        - filename – name of the output file to be read;
        - condition – declares if the pie chart if for condition 1 or 2.
    - Gets all the lines of the output file, in JSON format, for it to be built in form of a bar chart, different from the one produced above, in the browser by JavaScript and Highcharts library.

9. **/annotation/session/{sessionID}/getHeatmap**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Verifies whether an heatmap was produced or not. Returns this information in a string.

10. **/annotation/session/{sessionID}/getInputFiles**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Gets the names of the input files stored in the input folder.

11. **/annotation/session/{sessionID}/getIsomirsDETable**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Gets all the lines of the file, containing the differential expression table, in form of an array, for it to be built in the browser by JavaScript.

12. **/annotation/session/{sessionID}/getMirdeepTable**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Gets all the lines of the file, containing the novel miRNAs table, in form of an array, for it to be built in the browser by JavaScript.

13. **/annotation/session/{sessionID}/getPieChart/{filename}/{condition}**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
        - filename – name of the output file to be read;
        - condition – declares if the pie chart if for condition 1 or 2.
    - Gets all the lines of the output file, in form of an array, for it to be built in form of a pie chart in the browser by JavaScript and Highcharts library.

14. **/annotation/session/{sessionID}/StartDEAnalysis**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Invokes the script `deseq.pl` to start the detection of isomiRs.

15. **/annotation/session/{sessionID}/StartFindncRNAs/{mismatches}/{genomicHits}/{species}**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
        - mismatches – number of mismatches;
        - genomicHits – number of genomic hits;
        - species – species name;
    - Invokes the script `find_ncRNAs.pl` to start the alignment of sequences and detection of ncRNAs.

16. **/annotation/session/{sessionID}/StartIsomirAnalysis**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Invokes the script `find_isomiRs.pl` to start the detection of isomiRs.

17. **/annotation/session/{sessionID}/StartMirdeep**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Invokes the script `mirdeep.pl` to start the prediction of novel miRNAs.

18. **/functional/session/create**
    - GET Method;
    - Creates an entry in the database table `FunctionalForms` and also `Sessions`, but only if the user is performing functional analysis independently from annotation analysis. Returns the id to be used as session ID.

19. **/functional/session/{sessionID}/check/EndOfTargetPrediction**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Verifies if output files concerning target prediction in the pipeline are created and reads them. This information is returned to the browser for updating the progress bar.

20. **/functional/session/{sessionID}/getSelectedIsomiRs**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Gets the selected isomiRs selected in the differential expression table in the annotation analysis.

21. **/functional/session/{sessionID}/getTopGoTable**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;.
    - Gets all the lines of the output file, containing the top GO terms table, in form of an array, for it to be built in form of a pie chart in the browser by JavaScript and Highcharts library.

22. **/functional/session/{sessionID}/StartTargetPrediction**
    - GET Method;
    - Parameters:
        - sessionID – ID of the current analysis;
    - Invokes the script `functional.pl` to start the detection of isomiRs.

The pipeline scripts invoked by some of the above web services are represented in Figure 3.3.

3.3.3 High-level system behaviour

In Figure 3.7 and Figure 3.8 two diagrams are displayed with the interaction between the most relevant components of the system. The diagram in Figure 3.7 starts with the user defining the settings of the annotation analysis, which are stored in the database `AnnotationForms` table, being after redirected to a View (`annotation.progress`) that allows the user to visualize the progress of the analysis.

The arrows that point to the `AnnotationProgressController.php` correspond to web services that invoke functions within this controller.

The arrows with the following description: Start alignment and detection of ncRNAs, prediction of novel miRNAs, detection of isomiRs and diferential expression analysis, correspond to the web services 15, 17, 16 and 14 listed above, respectively. These services trigger the execution of several scripts in the pipeline layer.

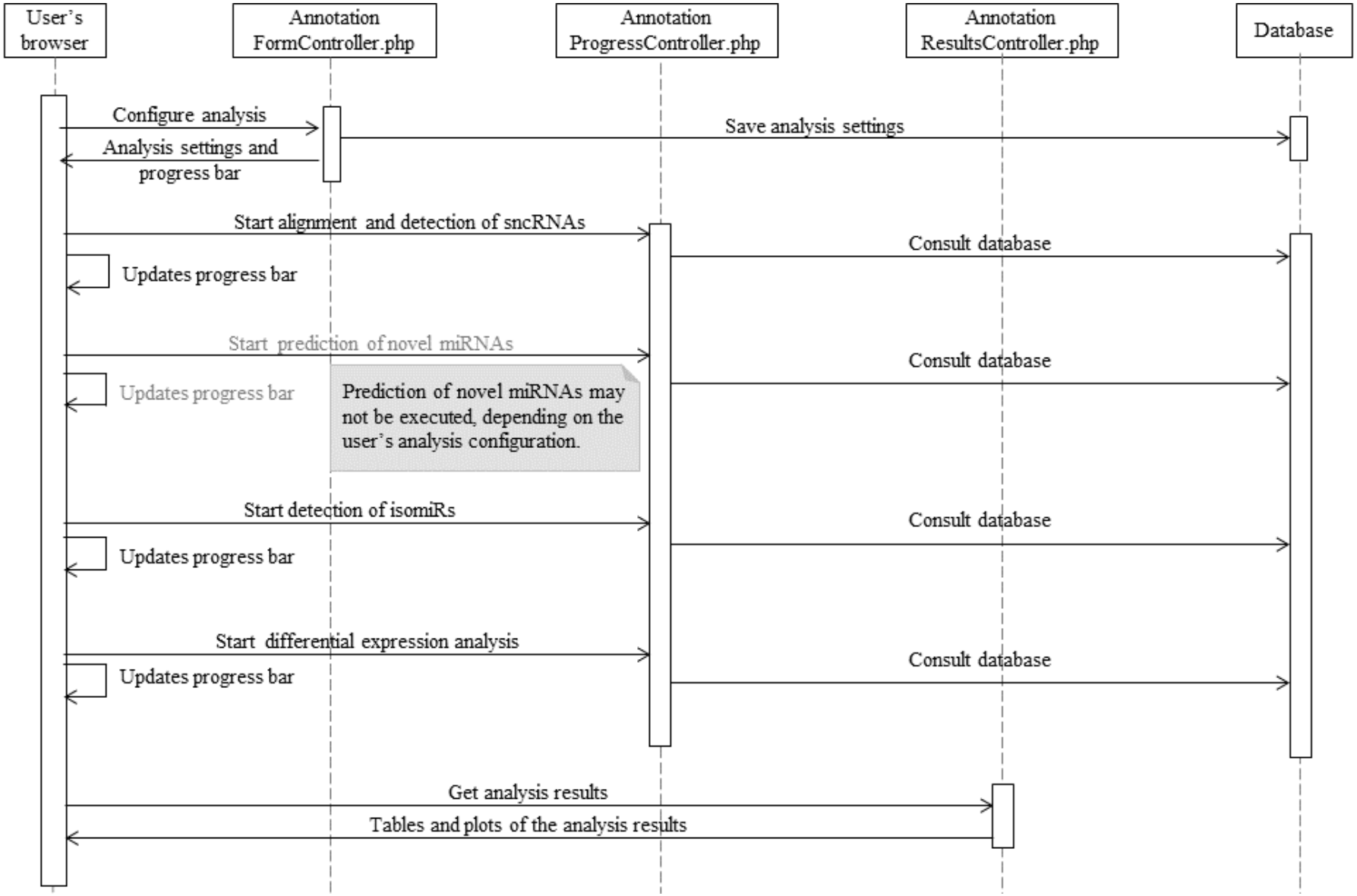


Figure 3.7. Annotation analysis sequence diagram.

Additionally, there are also web services designed to verify the state of the different phases of the pipeline. This allows the update of the progress bar present in the View, providing more information to the user (web services 2, 6, 5, 4 and 3). Finally, when the analysis is finished, it is possible for the user to proceed to a different View, `annotation.consultGeneral Results`, and posteriorly `annotation.consultResultsDE`, which displays the results produced by the scripts of the pipeline.

The diagram in Figure 3.8 reflects the general actions that occur when the user performs a functional analysis. It is possible to observe that there are less actions occurring in this case, when compared to the ones in Figure 3.7 that represents annotation analysis.

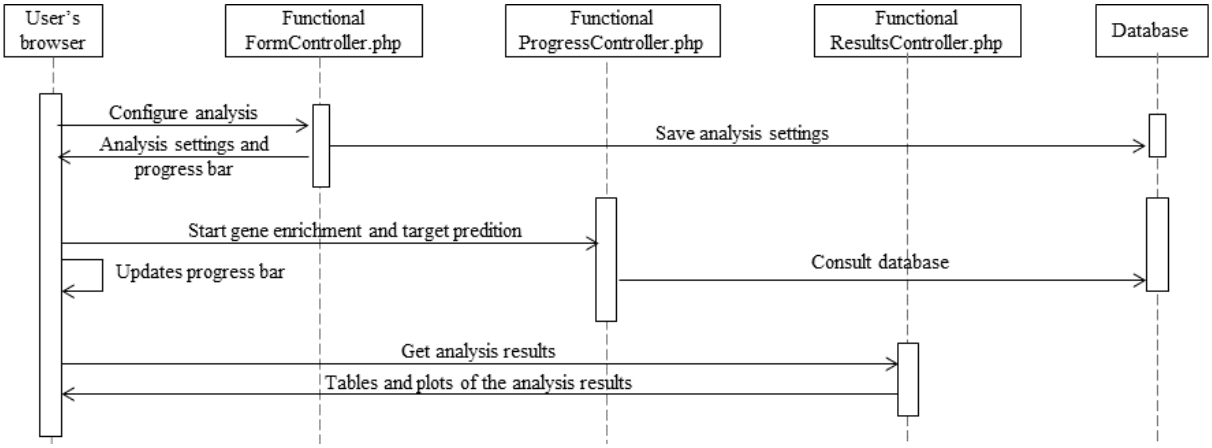


Figure 3.8. Functional analysis sequence diagram.

Similarly to annotation analysis, the user starts the analysis by defining the settings, which are stored in the database `FunctionalForms` table. After submitting this configuration, the user is redirected to the View `functional.progress`, different from the one mentioned in Figure 3.7. In functional analysis, `FunctionalProgressController` is responsible for triggering the execution of a pipeline script, after the web service number 22 invokes a function of this controller. This is represented by the arrow that contains the Start gene enrichment and target prediction description.

To verify the realization of the phases involved in this analysis, a web service, number 19, invokes a different function in the `FunctionalProgressController`. After the analysis is completed, the user may proceed to View `functional.consultResults` that displays the results concerning this analysis.

3.3.4 Interactions with the pipeline

The connection between the web application and the pipeline is represented in Figure 3.9, for annotation analysis, and in Figure 3.10, for functional analysis. The scenario simulated for these diagrams was the same as the one in Figure 3.4 and Figure 3.5, in which the user inserted two files for experimental condition 1 and one file for experimental condition 2.

The diagram shown in Figure 3.9 demonstrates which web services are called during the annotation analysis to initiate each task within it. Session ID is a parameter that is shared by all web services. The figure also shows which Perl script are invoked by each web service, along with its input and output files.

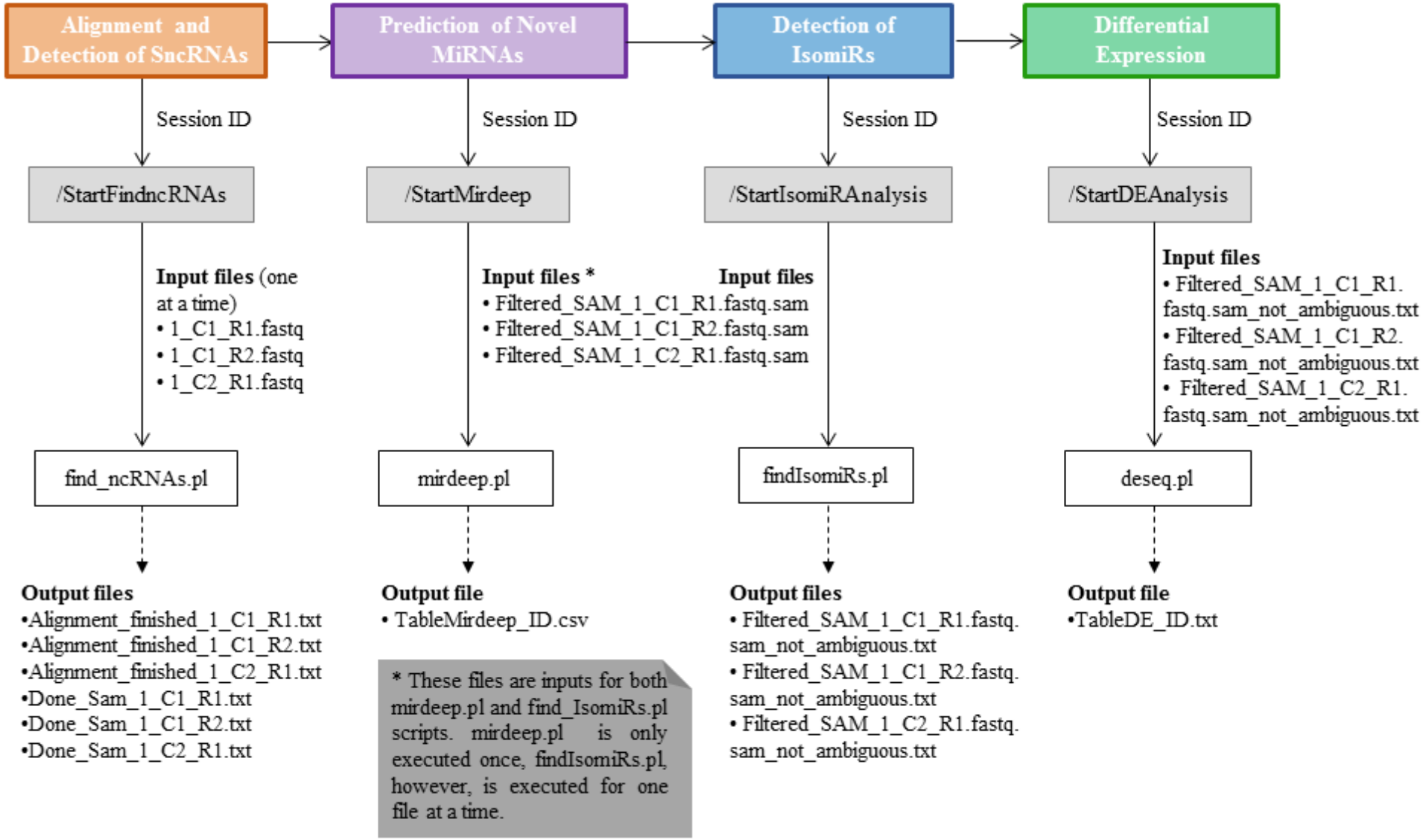


Figure 3.9. Relation between the phases of the annotation analysis, web services and the pipeline scripts. The arrows represent the data that is read or written. Web services are represented by grey boxes, colored boxes represent initiated tasks, Perl scripts are represented by white boxes.

The diagram is divided into four phases, matching the boxes named Alignment and Detection of sncRNAs, Prediction of Novel MiRNAs, Detection of IsomiRs, and Differential Expression. The output files displayed are the ones that affect the progress of the analysis, not being these represented in total. All the output files are shown in Figure 3.5.

The diagram concerning functional analysis, in Figure 3.10, is less complex than the one about the annotation analysis, having one phase named Target Prediction and Gene Enrichment. As in the diagram displayed above, only the output files that influence the progress of the analysis are represented. The remaining output files, and the input file in the figure below, are represented in Figure 3.4.

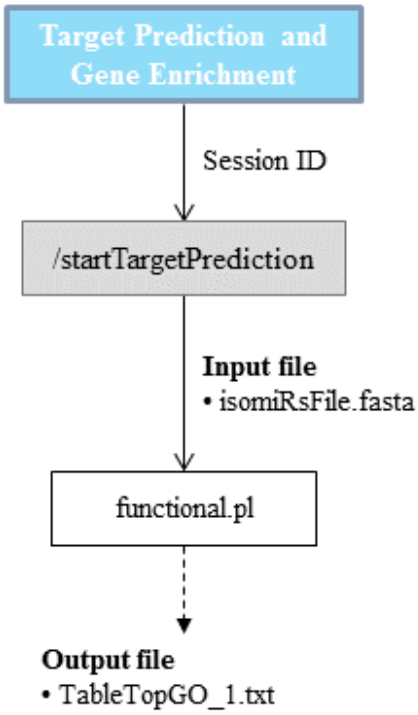


Figure 3.10. Relation between the phases of the functional analysis, web service and the pipeline script. The arrows represent the data that is read or written. Web services are represented by grey boxes, colored boxes represent initiated tasks, Perl scripts are represented by white boxes.

The web application starts the annotation analysis module by the verifying if the phases alignment and detection of sncRNAs have been initiated and, if so, terminated. This is represented in Figure 3.11. The first step is to check the end of the alignment phase. If the number of `Alignment_Finished` files, stored inside `1/AnnotationAnalysis/ReplicatesCondition1/ProcessingMessages`, visible in Figure 3.5, is the same as the number of the inputs files for experimental condition 1 inserted by the user, it means the alignment phase has ended with success. The same applies for the files of experimental condition 2.

If not, it is necessary to verify if the alignment phase is still running by confirming the existence of its corresponding control file (`1/AnnotationAnalysis/commonProcessingMessages/AlignmentStarted.txt`). Therefore, if this file exists, it means that the analysis has started but has not ended, and therefore should not restart. If no control file exists, two scenarios are possible. Whether the alignment phase has already ended but errors were encountered or it has not started.

In the scenario where the phase has ended unsuccessfully, the progress of entire annotation analysis stops and the error messages are displayed in the browser. If the analysis has not started, it must be initiated by the creation of the alignment control file, the same mentioned above. The creation of this

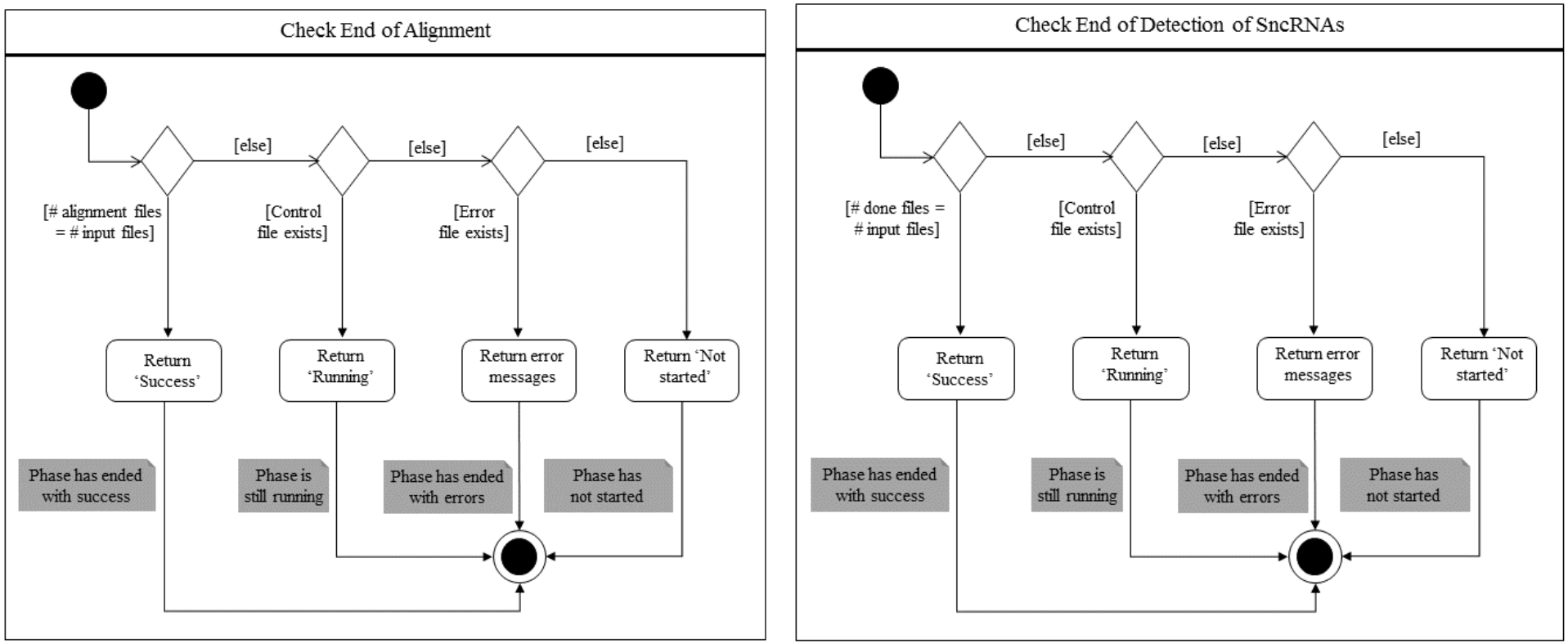


Figure 3.11. Activity diagrams for verification of the alignment and detection of sncRNAs phases of the annotation analysis module. The top black circle represents the start of the activity, and the bottom black circle the end of this activity. Rectangles represent actions, whereas diamond shapes represent decisions.

file is achieved when the Alignment and Detection of sncRNAs starts, which is visible in Figure 3.12. These two phases are coupled because they are run in the same Perl script, `find_ncRNAs.pl`, which is executed for each FASTQ file the user has inserted at the start of annotation analysis module.

When these executions end, it is necessary to delete the control file (`AlignmentStarted.txt`), the same for the two phases, meaning that the Alignment and Detection of sncRNAs has ended. After this, it is necessary to check if each phase has ended successfully, and for this reason the check of the end of the alignment is performed once again. If the outcome of this control check is equal to success, then it is necessary to verify if the second phase, detection of sncRNAs, also present in Figure 3.11, has also ended successfully. If the response is also equal to success, then the analysis advances to the prediction of novel miRNAs. If any of the phases has ended with errors, then these are passed to browser, informing the user of the problems encountered.

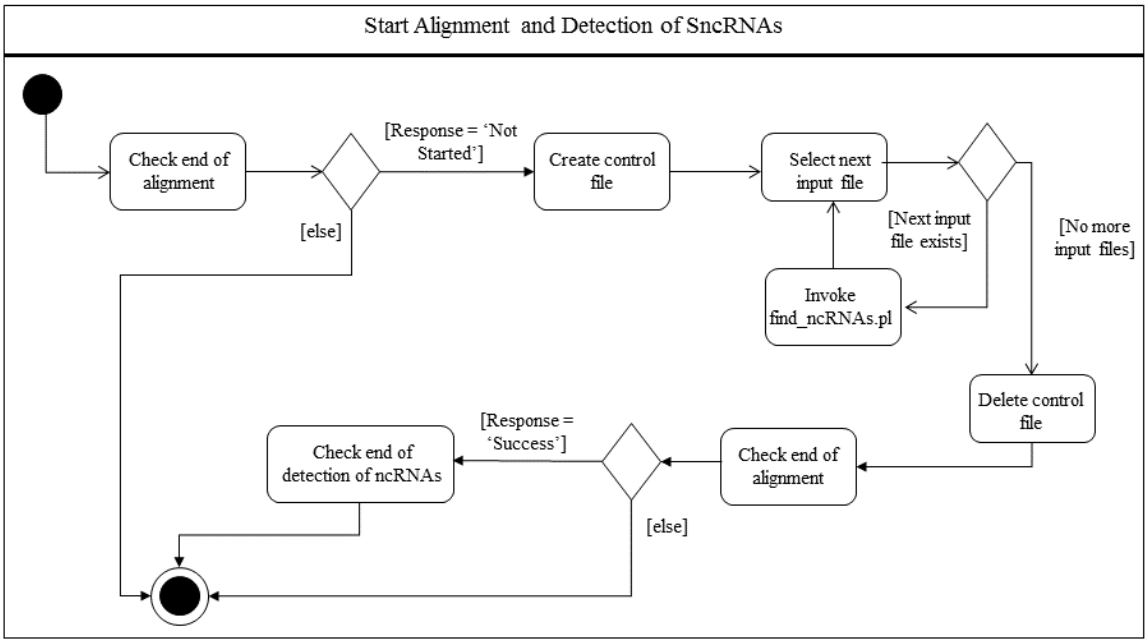


Figure 3.12. Activity diagram for the start of alignment and detection of sncRNAs phases for the annotation analysis module.

The next phase in the annotation analysis module is the prediction of novel miRNAs. This phase is optional and for this reason only starts if the user has chosen accordingly. If so, it is necessary to verify if the Mirdeep table (the last output produced in this phase) exists. If this is confirmed, then it is possible to know that the analysis has ended successfully. Otherwise, it is necessary to check the existence of the control file, `MirdeepStarted.txt`, for this phase (stored inside the `1/AnnotationAnalysis/commonProcessingMessages` directory). If the control file exists, the analysis is still running, if not, it is necessary to check this phase has been started but ended with error files. If no error file exists, it means this phase must be initiated. These verifications are displayed in Figure 3.13.

Like the start of every annotation analysis phase, a control file is created and, for this phase, the Perl script `find_mirdeep.pl` is invoked. This is performed once because this script receives all the input files, displayed in Figure 3.9, as arguments. Finally, the control file is deleted, representing the end of this phase. It is necessary to verify if there were errors, and this is represented in Figure 3.14.

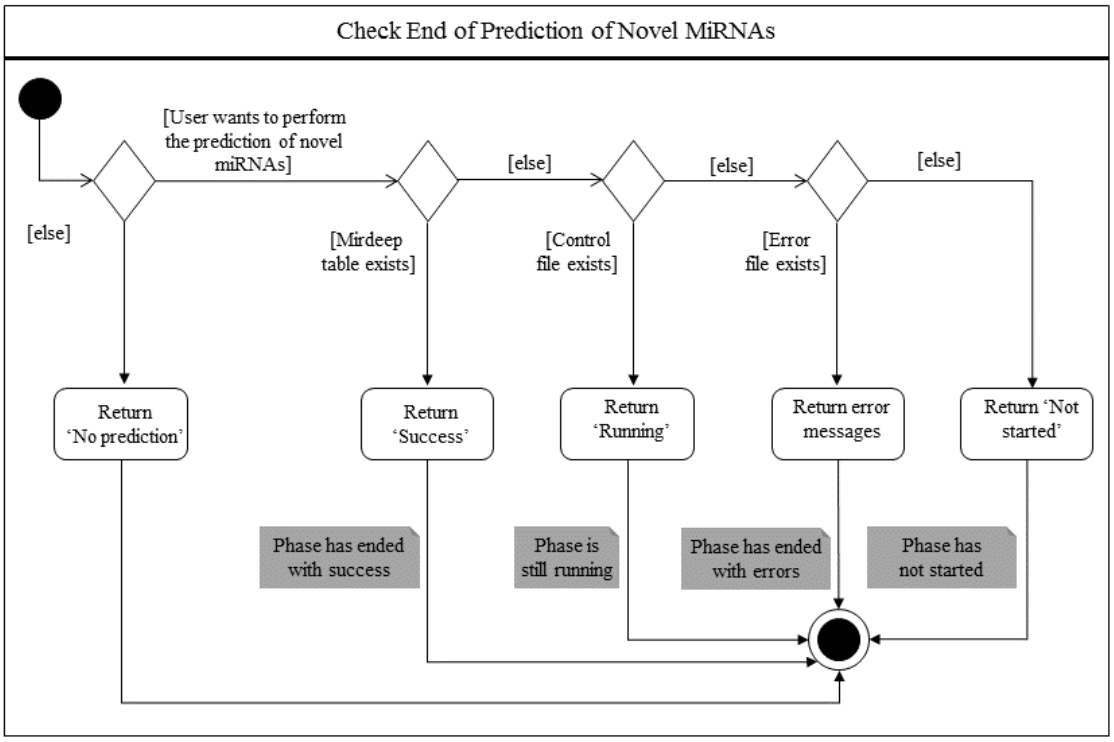


Figure 3.13. Activity diagram for the verification of the end of the prediction of novel miRNAs phase of the annotation analysis module.

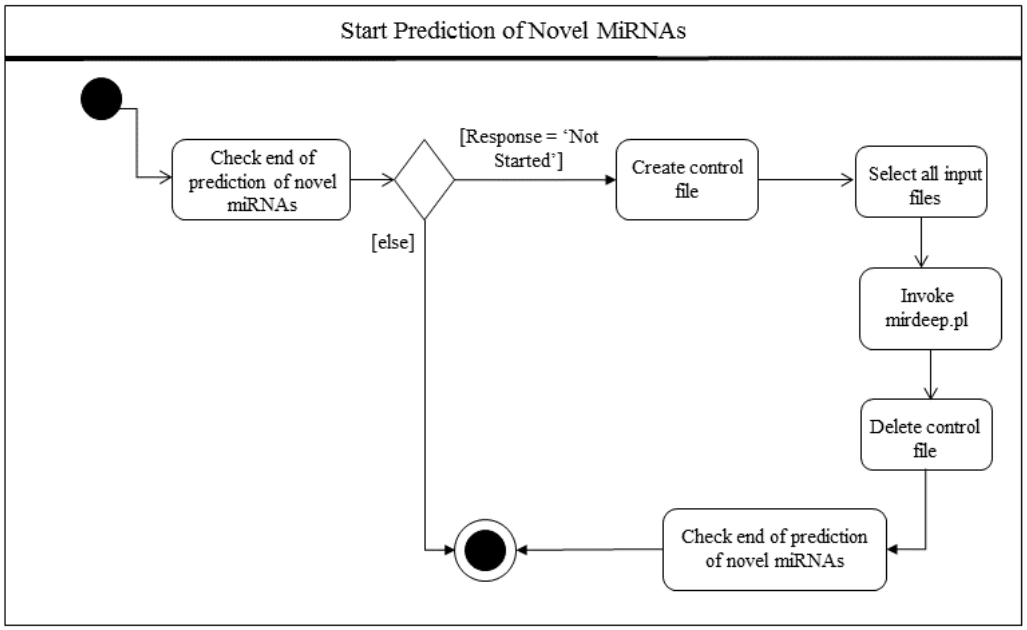


Figure 3.14. Activity diagram for the start of the prediction of novel miRNAs phase for the annotation analysis module.

In the detection of isomiRs phase, the processing is similar to previous steps shown in Figure 3.11 and Figure 3.13. Again, the progress of the analysis is conditioned by the accurate production of the find_isomiRs.pl output files (Figure 3.15).

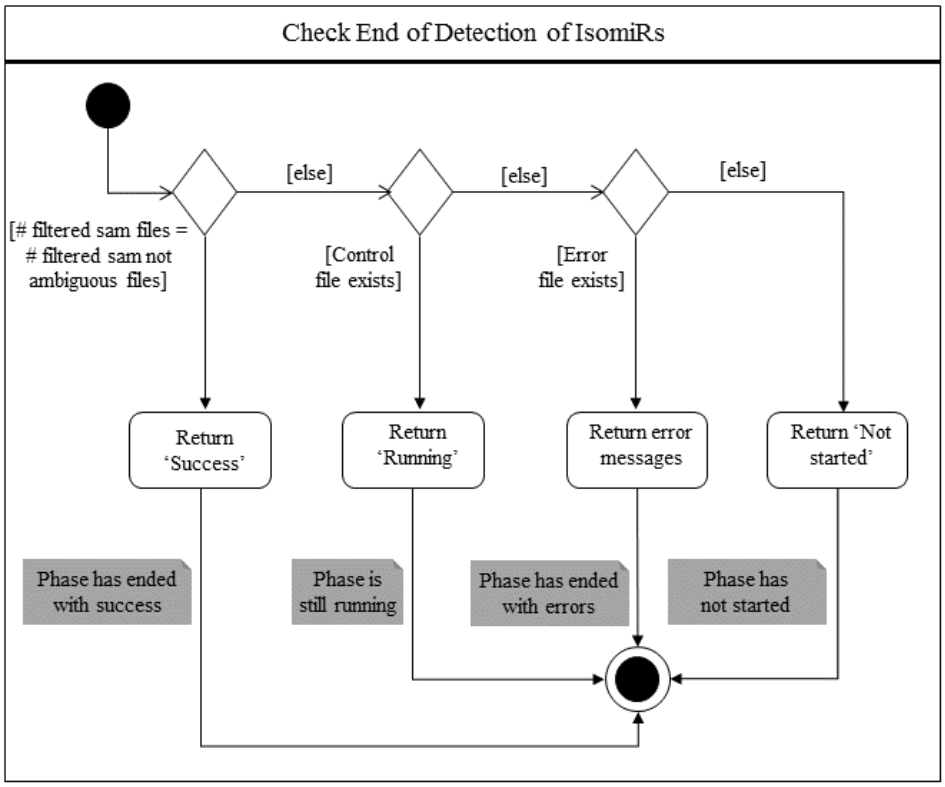


Figure 3.15. Activity diagram for the verification of end of the detection of isomiRs phase of the annotation analysis module.

If the analysis has not started, the control file is created (`IsomirsStarted.txt`), and the Perl script `find_isomiRs.pl` is invoked as many times as the number of input files, allowing processing in parallel several input files. After the control file is deleted, it is necessary to check the outcome of these executions, as displayed in Figure 3.16.

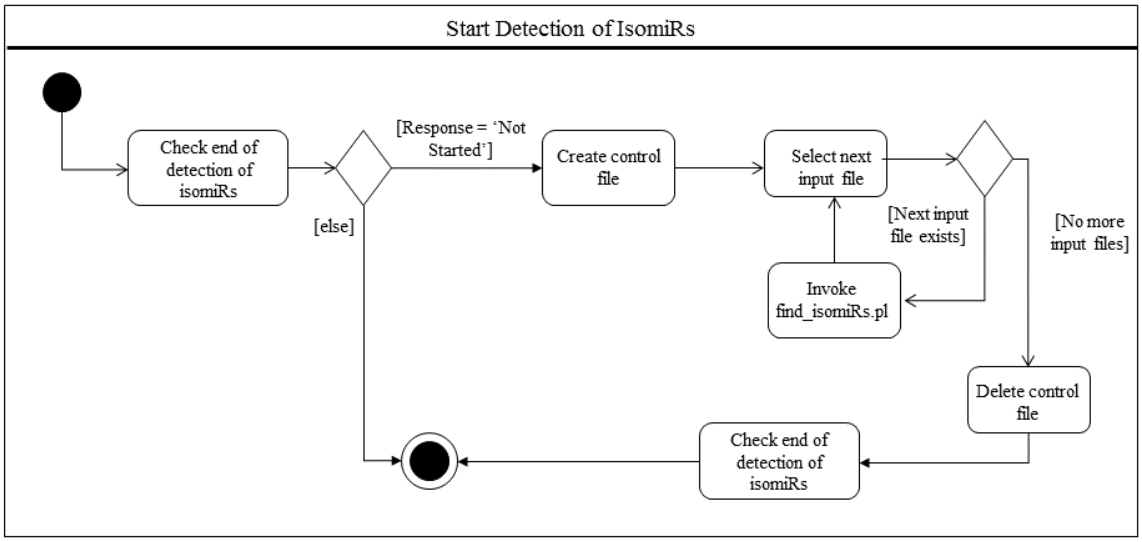


Figure 3.16. Activity diagram for the start of the detection of isomiRs of the annotation analysis module.

In the differential expression analysis phase, it is necessary to verify if the DE table exists. If true, then this phase has ended with success. As for the other phases, it is always necessary to check the existence of the control file and error files, in the case of the non-existence of the control file.

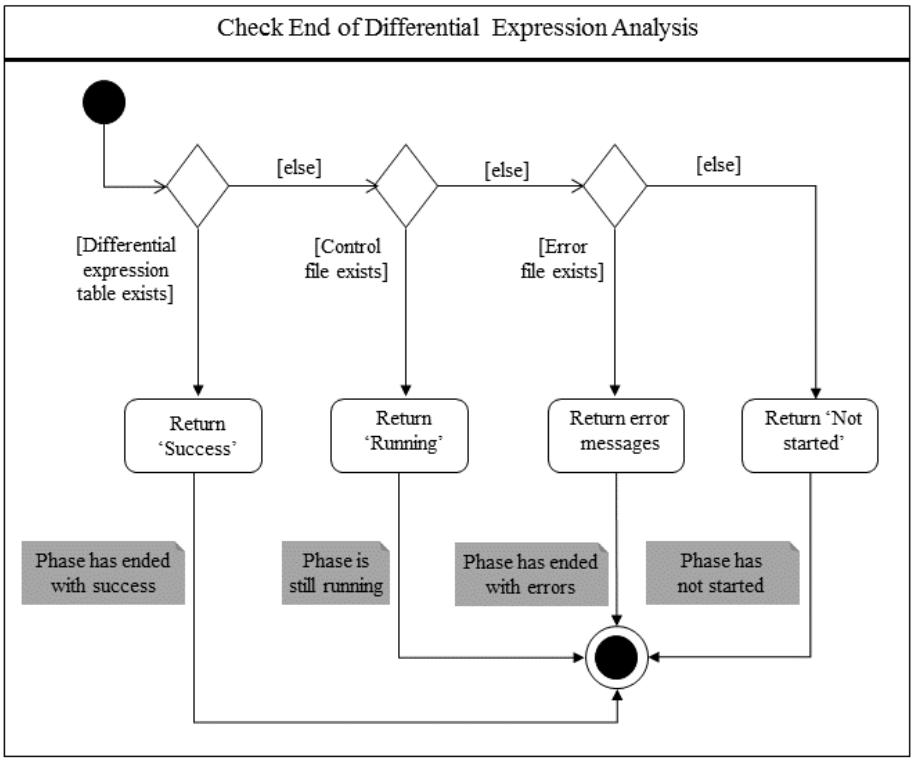


Figure 3.17. Activity diagram for the verification of the end of differential expression analysis phase of the annotation analysis module.

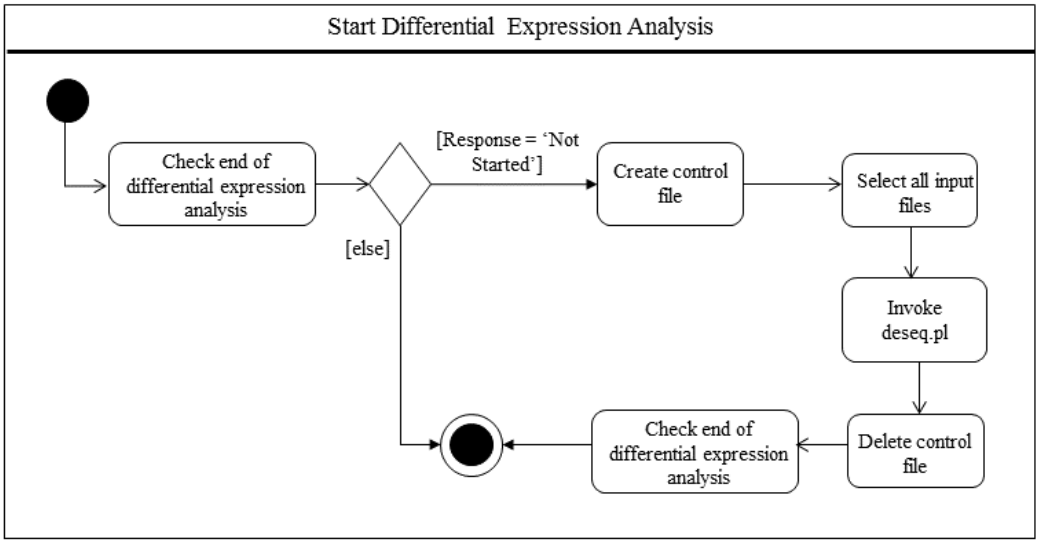


Figure 3.18. Activity diagram for the start of differential expression analysis of the annotation analysis module.

If this phase has not been performed, then the Perl script `deseq.pl` is invoked only once, receiving all input files, visible in Figure 3.9, as arguments. As for the other phases, it is necessary to verify the outcome of this one.

The `topGo` table symbolizes the end of target prediction and gene enrichment phases, and therefore if it does not exist, then other verifications must be performed. If it is concluded that these phases have not started, this must be performed. This is displayed in Figure 3.19 and Figure 3.20.

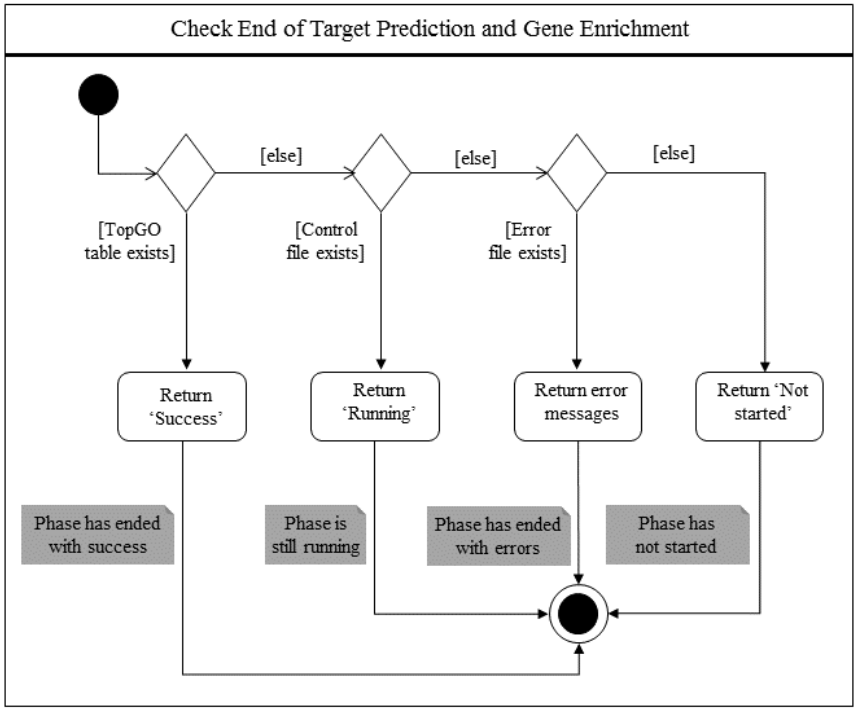


Figure 3.19. Activity diagram for the verification of the end of target prediction and gene enrichment phases of the functional analysis module.

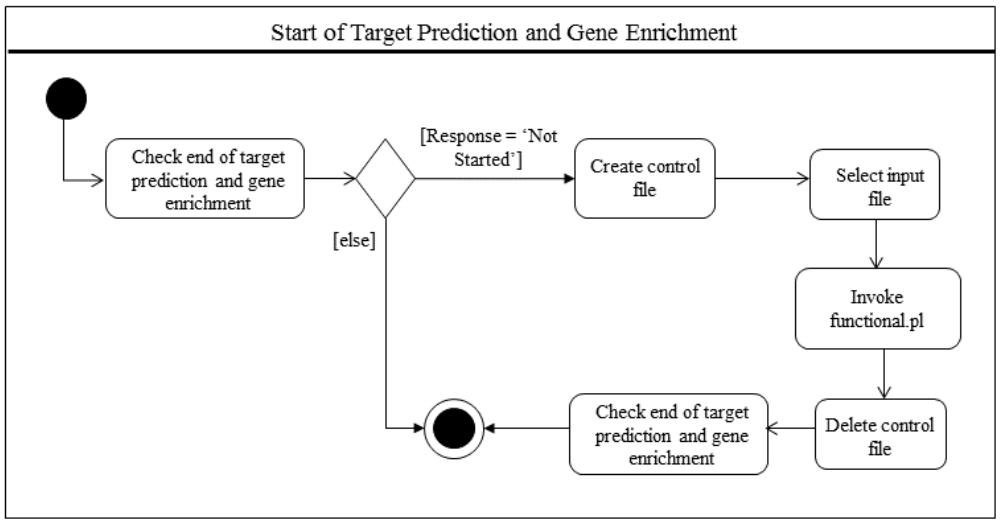


Figure 3.20. Activity diagram for the start of target prediction and gene enrichment of the functional analysis module.

The two phases, target prediction and gene enrichment, that compose the functional analysis module are performed by the same Perl script `functional.pl`. This script only receives one input, `isomiRs File.fasta`, visible in Figure 3.10.

3.4 Frontend

A web interface is a graphical frontend accessible via a web browser. An interface was built for the IsomiR Window tool, with instructions covering the several steps of the analyses, and with error messages that inform the user on how to solve errors, should they arise. The web application was tested on Google Chrome and Mozilla Firefox.

The designed web interface has three major areas: home, annotation analysis and functional analysis. In the sections below a description of the most relevant areas of the application is given.

3.4.1 Home

In the homepage (Figure 3.21), correspondent to focus area 1 of the UED presented in Figure 3.2, two main sections are considered the most relevant and they are the review and the start of an analysis.

Additionally, the homepage allows the user to navigate to sections such as: About the project, Help and Documentation and Contact us. Among these sections, Help and Documentation was designed to improve the user experience while navigating in the application. This section offers a description of each analysis and a guideline for the user to follow when performing the analyses.

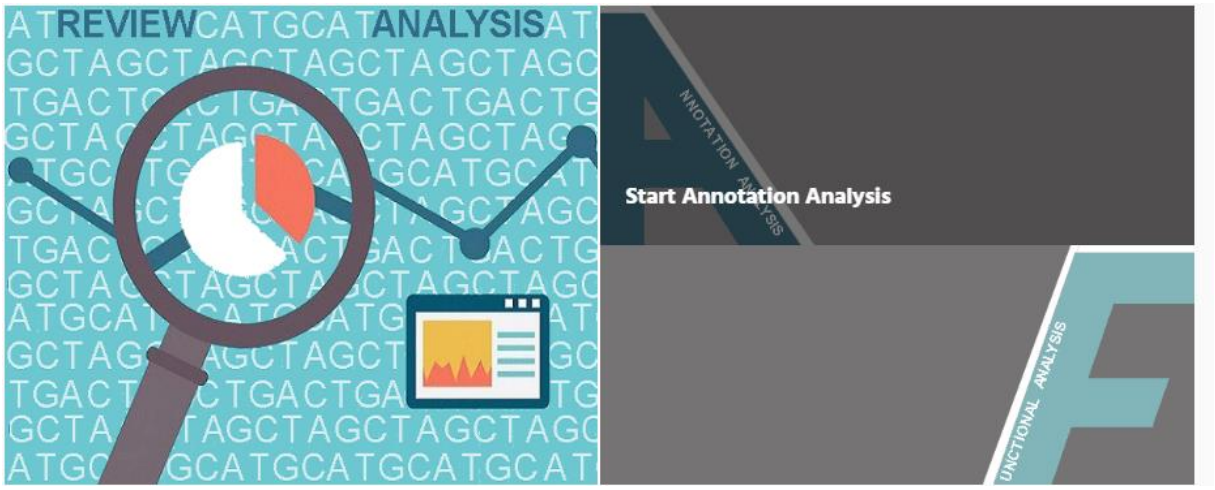


Figure 3.21. IsomiR Window homepage.

If the user chooses to start a new analysis it is necessary to select between the two possible analyses. This can be observed in Figure 3.21. The homepage also provides two links that allow the user to visualize the results for annotation and functional analysis performed in a demo session.

The next sections provide a description of the annotation and functional analysis and a description of how the user can review these analyses.

3.4.2 Annotation analysis

The first part of an annotation analysis is the definition of its configuration, which is displayed in Figure 3.22. This corresponds to focus area 2 of the UED presented in Figure 3.2. It is important to mention that when the user is in the context of performing annotation analysis (or functional analysis), a side bar is available to allow the user to switch between analyses or to get help concerning the analysis.

The first step of the annotation analysis is the upload of data files in FASTQ format (see Section 2.1.2). The user must upload at least one file per experimental condition. The tool allows the user to choose between two experimental designs, relative to paired or unpaired samples. If the user chooses to perform an analysis with paired samples, more than two files, instead of one, per experimental condition

must be uploaded, due to third-party software packages requirements. An example of the files for each condition is provided.

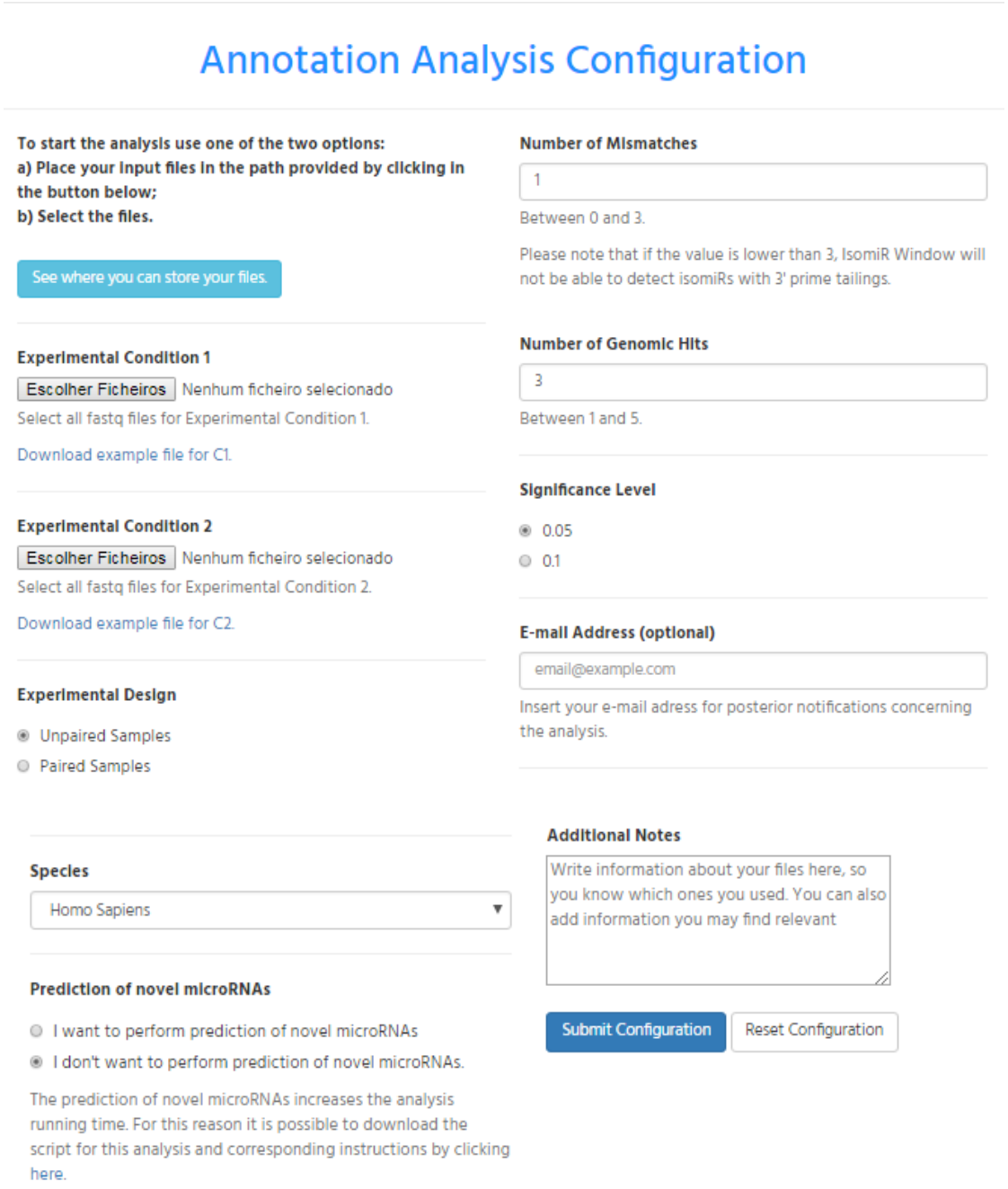


Figure 3.22. Annotation analysis configuration webpage.

The file upload can be done by one of two options:

- Upload the files in the browser. This option is easier because the user must only click in select files in the experimental condition sections and chose the ones for analysis.
- Manually insert the files in a path in the local filesystem. In this case, the user must click on a button that says “See where you can store your files”. After this, a path is displayed in the webpage, where the user must insert the files. With this option the files are uploaded faster but it might be limited to researchers with low proficiency in Linux environments.

Some of the parameters in the form in Figure 3.22 have default values (recommended) but the user can change them, within the possibilities. The first is the Species field, which is a drop-down list with the names of species from which the samples originated. Only one species for analysis can be chosen. For now, due to species data availability constraints (which limits the functioning of the pipeline), only one species, *Homo sapiens* is available for selection in the Isomir Window application interface. However, the application is ready to accept other species, namely *Mus musculus*, *Drosophila melanogaster*, *Sus scrofa*, *Bos taurus*, *Gallus gallus*, *Capra hircus*, *Equus caballus*, *Ovis aries*, *Canis familiaris*, *Danio rerio*, *Pan troglodytes*, and *Rattus norvegicus*.

In case the user wishes to perform novel miRNA prediction, the Prediction of novel microRNAs field must be selected. The number of mismatches must have values between 0 and 3 and the number of genomic hits is between 1 and 5. All these fields are used as arguments to be passed to the Perl scripts. The e-mail address input field is optional, but if filled, allows the user to receive e-mails reporting to the progress of the analysis. An e-mail is sent when the analysis starts, which also contains the information relative to the chosen configuration, when it ends, and whenever an error is encountered. The Additional Notes field allows the user to write information concerning the uploaded files or any other information relevant for characterizing the context of the analysis.

Each field is verified using JavaScript functions and whenever there is an invalid value, a red color appears in the field indicating the problem. If the user does not correct the values in the fields, then an error message is received when submitting the configuration. The analysis only starts when there are no remaining errors.

After submitting the configuration, the user is redirected to a web page that displays a progress bar that is automatically updated, which is possible through AJAX technology.

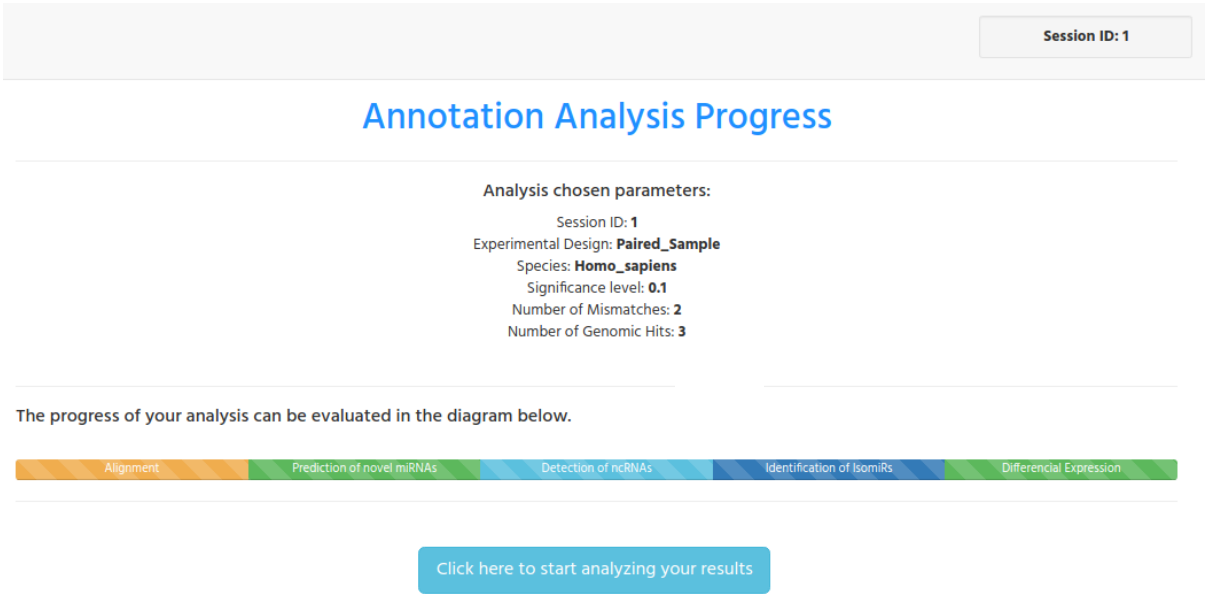


Figure 3.23. Annotation analysis progress webpage.

The main steps of the progress bar are in accordance with the Perl scripts that perform data processing. For the annotation analysis, there is a total of 4 scripts that are called sequentially (see Figure 3.9 and Figure 3.11–Figure 3.18), and whenever a new script is called, the user is informed through the progress bar. This can be observed in Figure 3.23.

If an error occurs during the analysis, the user receives a message that appears in the browser, and additionally, if an e-mail address was provided, by e-mail. With the information given, the user must

correct the errors in the data files, if applicable, and restart the analysis. An example of an error message is presented in Figure 3.24.

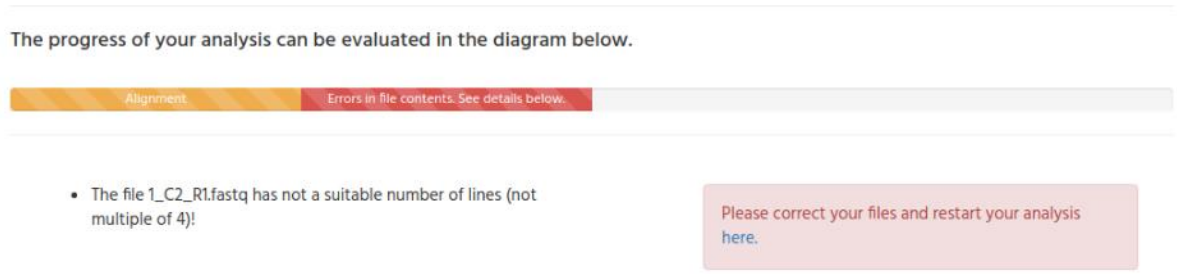


Figure 3.24. Error encountered during the alignment phase of the annotation analysis module.

If the analysis is performed without problems or warnings, a button is displayed (visible in Figure 3.23), allowing the user to proceed to a web page with the General Results. There is also a page of results of the annotation analysis that provides the visualization of the differential expression testing results. Both results web pages are described in Chapter 4.

3.4.3 Functional analysis

The user has two options to run the functional analysis: start the functional analysis based upon the results of the previous annotation analysis, or a new session can be created and configured. Functional analysis configuration interfaces correspond to focus area 5 of the UED represented in Figure 3.2.

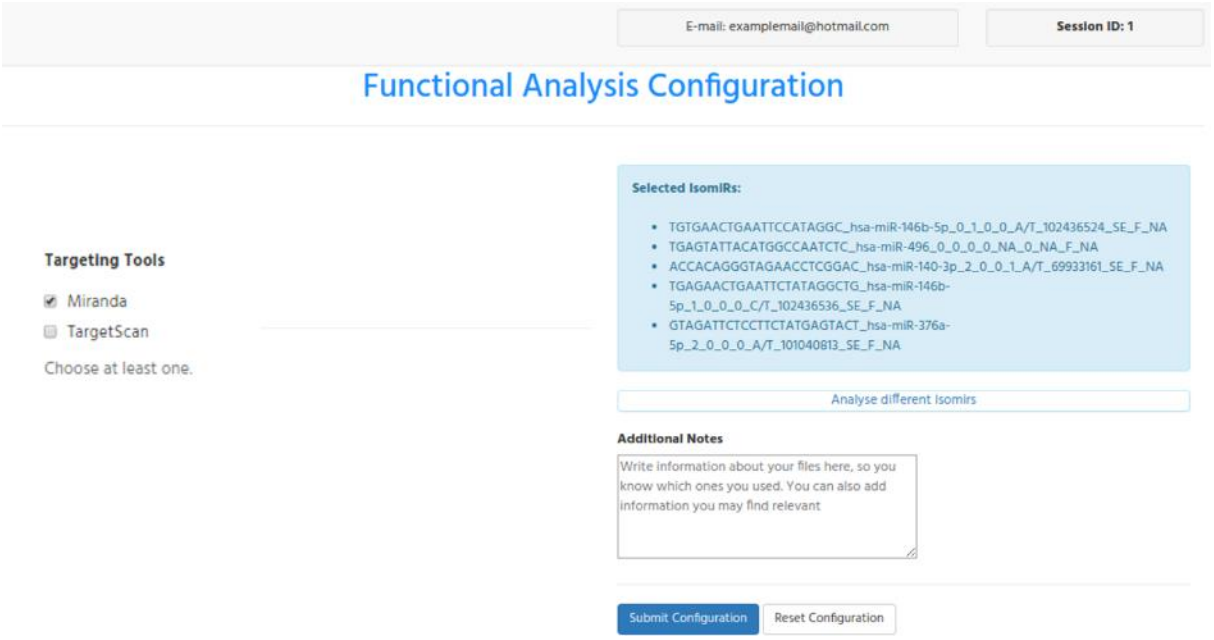


Figure 3.25. Functional analysis configuration with isomiRs found in annotation analysis.

In the first case, the user must select the isomiRs from the ones presented in the differential expression results page. Additionally, because the user already selected the species in the annotation analysis configuration form, this field does not appear as an option for the user. The Targeting Tools field allows the user to choose one or more tools to be used during this analysis. It is important to mention that the user has the possibility to perform the functional analysis on different isomiRs than the

ones chosen initially, by clicking in the button “Analyze different IsomiRs”. This scenario is illustrated in Figure 3.25.

If the user chooses to perform the functional analysis independently from the annotation analysis module, the interface in Figure 3.26 is presented. In this case, the user must submit only one data file in the FASTA format (an example file is provided), otherwise an error message is displayed. The analysis window also requires the input of the species being analyzed. Here, the options for file upload are equal to the ones presented for annotation analysis. The input field is also automatically verified in JavaScript and if an e-mail address is inserted, the user receives notifications regarding the analysis, as it happens in the annotation analysis.

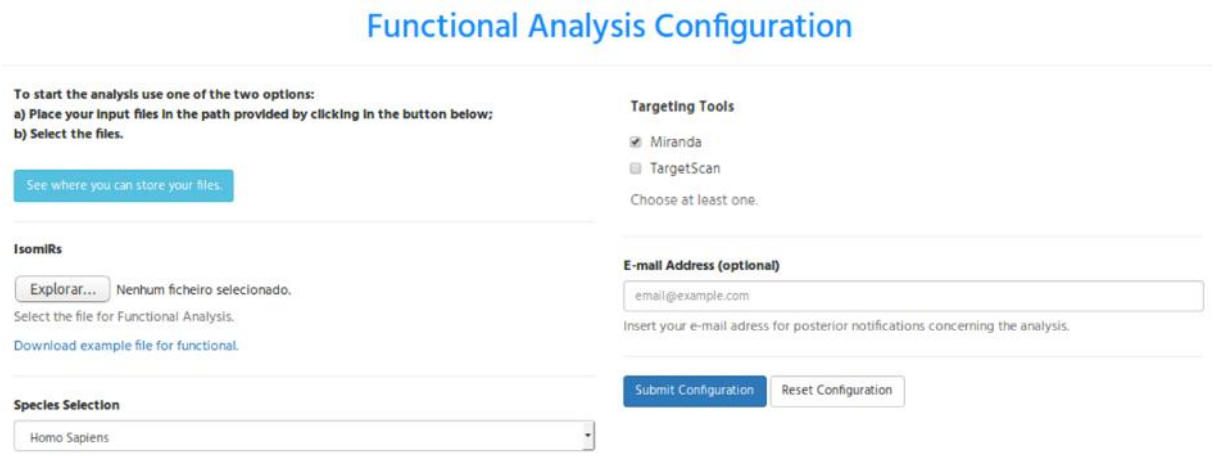


Figure 3.26. Functional analysis configuration webpage.

After the submission of the functional analysis configuration form, the user is redirected to a web page similar to the one presented in Figure 3.23, but with different tasks in each step of the progress bar, namely target prediction and gene enrichment.

3.4.4 Review analysis

In the homepage, besides starting one of the analysis modules, the user is given the option to review the results of a previous analysis. The review allows to select whether to visualize the results of the annotation analysis or the results of the functional analysis. This can be achieved through the insertion of the session ID in the input box, with the additional selection of the desired analysis, and this is demonstrated in Figure 3.27.

Several situations can occur when the user chooses to review an analysis. If the inserted session ID is not in the database, the user gets a message stating that no analysis was performed for that inserted session ID. Another situation is if the user desires to see the results of an analysis, but this analysis is not yet completed, and therefore the results are not available. In this case, the user is redirected to a web page that displays the progress of the analysis, through a progress bar, as displayed in Figure 3.23.

Figure 3.27. Review Analysis section of Home webpage

Another scenario is the user inserting a session ID and choosing to visualize the results of an analysis that was not performed. For example, the user only performed annotation analysis and did not continue to functional analysis. If this user chooses the functional analysis in the Review analysis form, a message is displayed, explaining that the annotation analysis was performed but no functional analysis was found.

As the final scenario, in which the user performed both analyses, no errors were found, and all the results are available, the user must always choose which one of the results to review first. If the first results to be seen are the ones relative to the annotation analysis, then the user is redirect to the web page with those results. The interfaces of these web pages are not displayed in this chapter since they are present in Chapter 4.

3.5 Summary

In this chapter, the web application for the IsomiR Window tool was described, with an additional summary on the other component of the tool, the pipeline. The application design and structure was also presented. The components of the backend, database and web services were described. The most relevant interfaces of the frontend were also displayed, always accompanied with a description of the functionalities available.

In the following chapter human datasets of small-RNA-seq were used to benchmark IsomiR Window tool.

Chapter 4 – Benchmarking the IsomiR Window tool

The development of IsomiR Window tool is only complete with the evaluation of its accuracy by testing with NGS datasets that correspond to data generated experimentally and holding at least 6 million reads in each library. For this purpose, some of the libraries generated in a study that investigated the effect of HIV-1 infection in the miRNA profile of human T [20] were selected. The study found a consistent down-regulation of miR34c-5p upon HIV-1 infection of stimulated T naïve T cells. However, changes at the level of isomiR expression were not analyzed or considered, providing an opportunity to benchmark the IsomiR Window tool.

4.1 Methods

In this section a description of the datasets used to benchmark the IsomiR Window tool is provided, as well as the settings defined for annotation and functional analysis.

4.1.1 Datasets

The datasets used for the analysis with the IsomiR Window tool were generated from human naïve stimulated CD4+ T cells obtained from healthy donors, which were assayed in the presence or absence of HIV-1 infection [20], and that are publicly available at Array Express database within accession E-MTAB-4214. The following libraries were selected, corresponding to stimulated naïve human CD4+ T cells non-infected (sample 4-ERS1473549 and sample 5-ERS1473550 respectively) and libraries that correspond to HIV-1 infected stimulated naïve human CD4+ T cells (sample 6- ERS1473551 and sample 7-ERS1473552). This data was generated with a paired sample design, which means that, from the same individuals, libraries of infected and non-infected conditions were created (sample 4 pairs with sample 6 and sample 5 pairs with sample 7).

4.1.2 Analysis settings

As described in the study [20], adapters were removed by searching for adapter sequence before sequence analysis.

In the beginning of the annotation analysis, several parameters values were established. The indication that the samples were paired was provided, along with the species (*Homo sapiens*), desired significance level for differential expression analysis (0.1), the number of maximum allowed mismatches in one read (2) and the number of maximum genomic hits of a sequence (3). These two parameters filter the sequences in the alignment task.

For the functional analysis, it was only necessary to define which targeting tools were to be used for target prediction, and both were selected, Miranda and TargetScan. The species was already chosen from the annotation analysis.

4.2 Results and discussion

In this section the results produced by the IsomiR Window tool are displayed and discussed. These concern sample quality control, miRNA complexity, differential expression of isomiRs and their functional impact.

4.2.1 Quality approval of datasets

The first chart available in the results, concerning the annotation of sequences, refers to length distribution of the reads found for each condition. Here, the user can use the scroll bar to navigate through the results found. Despite not being shown in the figure, there were reads/sequences with 15 to 44 nucleotides. Because read lengths of miRNAs are expected to peak between 22 and 23 nucleotides [83], the part of the bar charts displaying these results is presented in Figure 4.1. As desired, the results are consistent across the studied conditions, being that Condition 1 corresponds to libraries of non-infected stimulated naïve human CD4+ T cells and Condition 2 corresponds to libraries of HIV-1 infected stimulated naïve human CD4+ T cells.

Figure 4.1. Read length distribution. Bars represent the total number of raw read counts found for each read length for each replicate. Condition 1 (C1) corresponds to libraries of non-infected stimulated naïve human CD4+ T cells and Condition 2 (C2) corresponds to libraries of HIV-1 infected stimulated naïve human CD4+ T cells.

The graphics displayed in Figure 4.2, present the distribution of reads across the different types of sncRNAs in both conditions, showing the corresponding percentages. In this graphic, for a quality approval, it is expected that, at least, 70% of the reads correspond to miRNAs [20]. This is in accordance with what was found in the original study, being the small differences in the values due to the different versions of miRbase [46]. Only a small percentage, ≈2-4%, should correspond to rRNA, since a larger percentage might mean RNA degradation, and therefore the isomiR analysis will not retrieve highly reliable results (although degradation first affects longer RNA molecules). It is possible to observe that biological context is very similar between the two conditions.

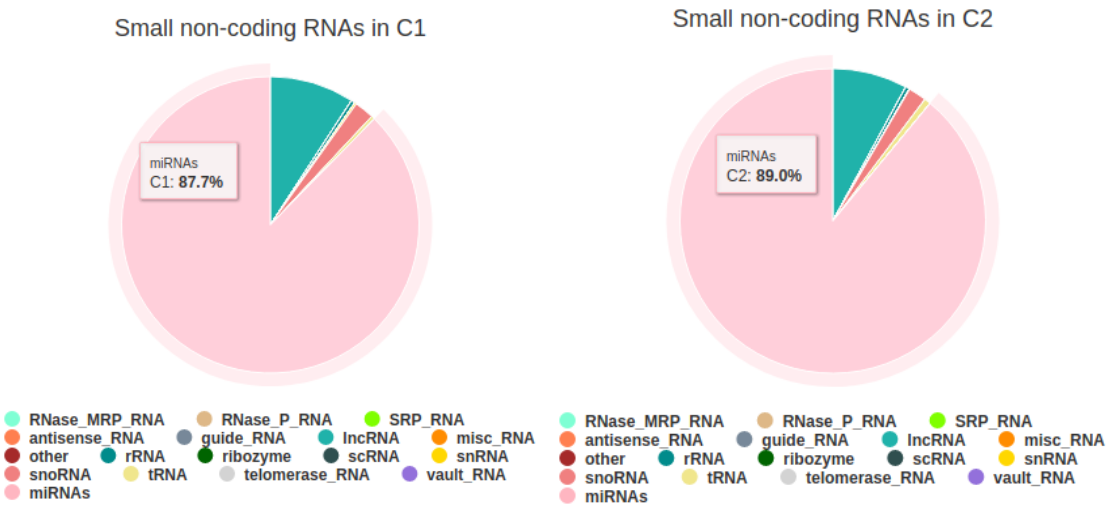


Figure 4.2. Distribution of reads across small non-coding RNAs types. The pie charts display the average percentage of reads for each sncRNA category found across all biological replicates in condition 1 (C1) and in condition 2 (C2).

Has shown in Figure 4.2, we observe in both studied conditions that miRNAs comprise the highest frequency of annotated reads, with 87.7% for condition 1, and 89.0% for condition 2. We can also observe that lncRNA (long non-coding RNA), snoRNA (small nucleolar RNA) and tRNA (transfer RNA) are after miRNAs, the most abundant sncRNAs in these samples.

The next result displayed by the tool allows assessing miRNA biogenesis, allowing to quantify from which arm of the pre-miRNA the obtained miRNA reads derive. In Figure 4.3 it is possible to observe that, for both conditions, the strand of the miRNA duplex (mature miRNA) from which the miRNAs originated was the 5p-arm strand, with 84.0% for condition 1, and 86.9% for condition 2. These graphics also demonstrate the quality of the datasets, once as reported in many publications, although NGS studies have shown that both arms of the pre-miRNA are able to produce miRNAs [46], the expression of the 5p-arm derived miRNAs is more often detected.

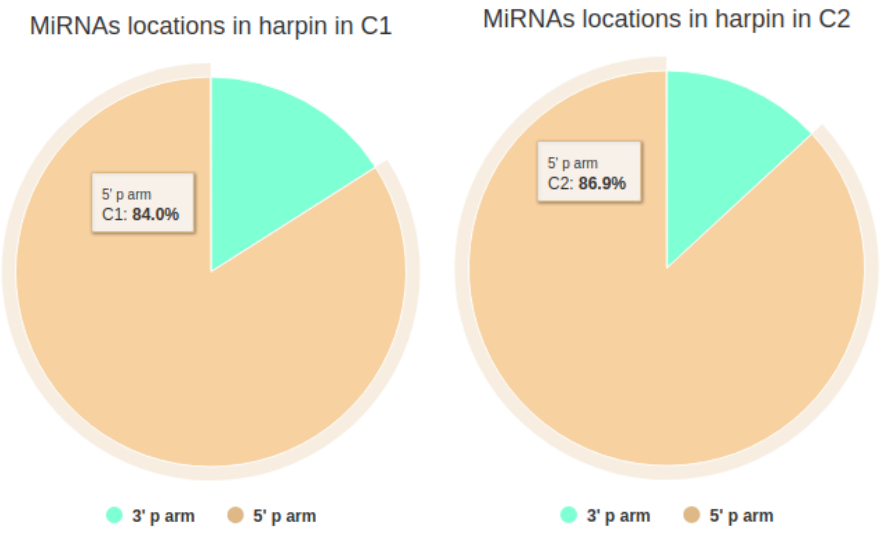


Figure 4.3. Frequency of miRNAs according to their biogenesis. This pie chart shows the average proportion observed in the biological replicates of each condition of 5p-arm and of 3p-arm derived miRNAs.

4.2.2 Unravelling the miRNA complexity

The results presented in Figure 4.4 display the miRNA raw counts found in each condition. In the figure are only presented the miRNAs more predominant in each condition, but is possible to get more information just by clicking in the scroll bar present on the right of each graphic. It is possible to see that miRNA-21 is the most predominant miRNA in both conditions, and has almost the double of counts in condition 1. miRNA-146 and miRNA-30 are next most predominant in the samples for both conditions. After these, the results for each condition vary a little and the counts are about 150k.

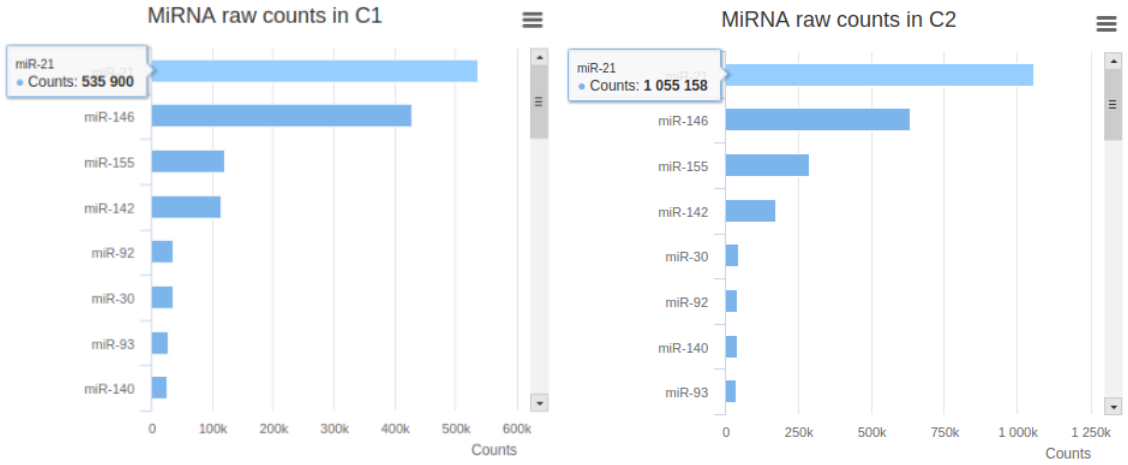


Figure 4.4. MiRNA raw counts in experimental condition 1 (C1) and 2 (C2). The bars display the average raw read counts found across replicates of each of the studied experimental conditions.

The IsomiR Window web application provides a zip file, containing all the data files from which all the graphics displayed in this chapter were based. One of these files holds all miRNA counts, which can be useful for the researcher to perform tests and investigate differences at the miRNA level.

The next result, presented in Figure 4.5, informs the user about the types of isomiRs found in each condition and their abundance. The IsomiR Window tool allows the identification of all types of isomiRs and also the combinations between all types of isomiRs. It is possible to see that 3' isomiRs are the most predominant in the samples, as expected [39], and the majority of them originate from 3' additions in the canonical miRNA, 77.7% for condition 1 and 76.3% for condition 2. The second most predominant type of isomiR, also an 3' isomiR, originates from trimming at the 3' end of the canonical miRNA. The other types of isomiRs present in the samples correspond to 5' end trimming, 5' end addition and 5' end trimming and 3' end addition. The results are similar for both experimental conditions.

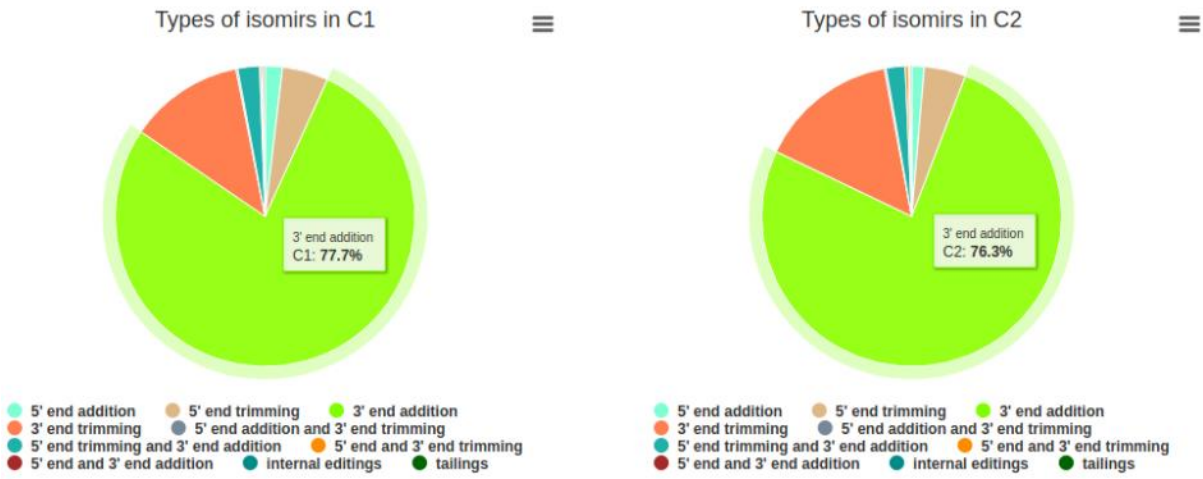


Figure 4.5. Different types of isomiRs found in experimental condition 1 (C1) and 2 (C2).

The IsomiR Window tool also displays the main editing events found for the samples of each condition. However, Figure 4.6 shows that none of these events were detected in the analyzed dataset.

Figure 4.6. IsomiRs editing events in experimental condition 1 (C1) and 2 (C2).

The IsomiR Window tool performs the prediction of novel miRNAs, if chosen by the user, which was the case for this analysis. If any novel miRNAs are encountered, then a table displaying these miRNAs and corresponding statistical measures is produced. However, for this dataset, novel miRNAs were not detected.

4.2.3 Differential expression of isomiRs

The results concerning differential expression are in a different web page than the results presented above. The user can visualize a heatmap, but only if there is differential expression of the isomiRs in the samples. Otherwise, a table is provided containing 100 isomiRs ordered by adjusted p-value.

With a level of significance <0.1, several isomiRs were found to be differentially expressed. For this reason, a heatmap was created (see Figure 4.7), displaying these differentially expressed isomiRs. One of the control samples (non-infected stimulated naïve CD4+ T cells) is clustering with the infected samples (HIV-1 infected stimulated naïve CD4+ T cells), which is not in accordance with the published study. However, in the published study, more samples from condition 1 were included and importantly the analysis was based on miRNA level, whereas the IsomiR Window tool performs the analysis on isomiR level.

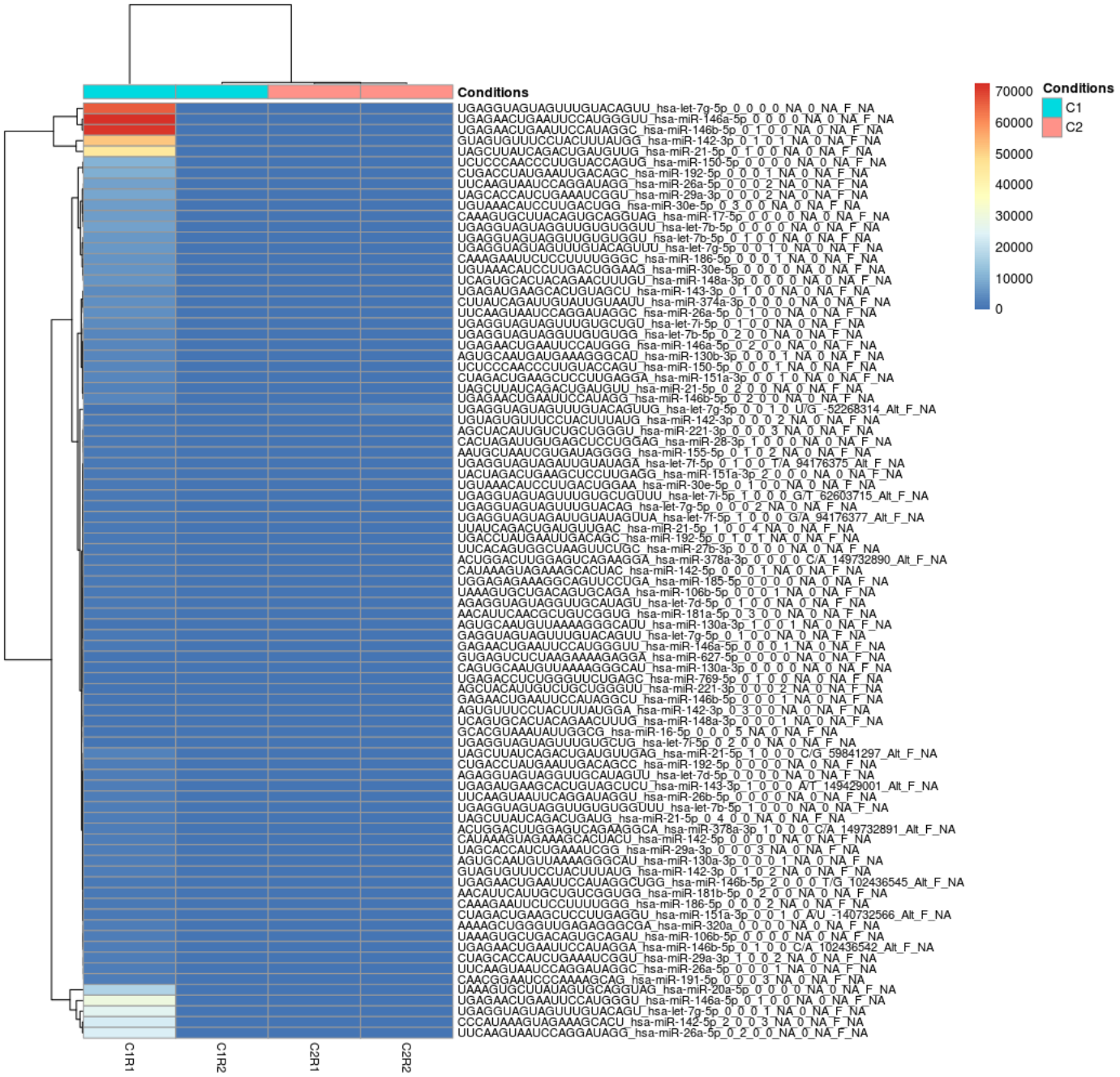
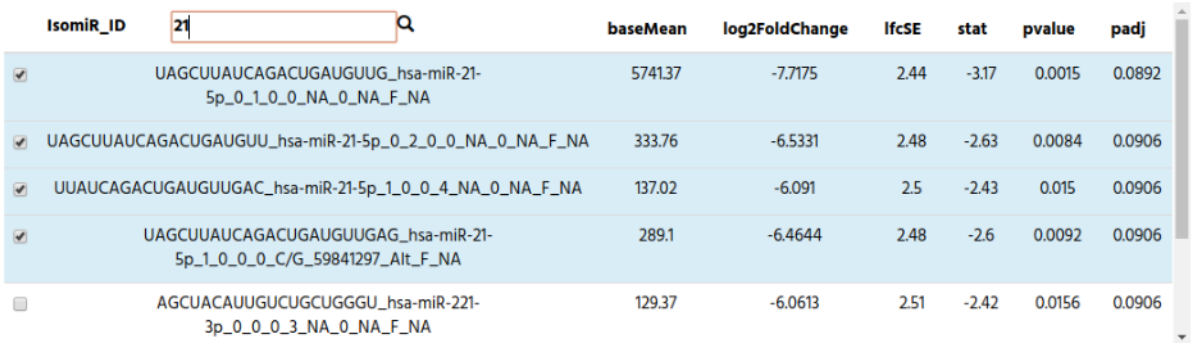


Figure 4.7. Heatmap displaying the differentially expressed isomiRs.

The table of differentially expressed isomiRs, shown in Figure 4.8, has 7 columns: the isomiR_ID, baseMean (average of the normalized count values, dividing by size factors, taken over all samples), the log2FoldChange (effect size estimate, describes how much the gene's expression have changed due to treatment with in comparison to control), lfcSE (standard error estimate for the log2 fold change estimate), stat (uncertainty of a particular effect size estimate as the result of a statistical test), pvalue (result of this test) and padj (applies the Benjamini-Hochberg adjustment to the p-value).



IsomiR_ID	21	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
☑	UAGCUUAUCAGACUGAUGUUG_hsa-miR-21-5p_0_1_0_0_NA_0_NA_F_NA	5741.37	-7.7175	2.44	-3.17	0.0015	0.0892
☑	UAGCUUAUCAGACUGAUGUU_hsa-miR-21-5p_0_2_0_0_NA_0_NA_F_NA	333.76	-6.5331	2.48	-2.63	0.0084	0.0906
☑	UUAUCAGACUGAUGUUGAC_hsa-miR-21-5p_1_0_0_4_NA_0_NA_F_NA	137.02	-6.091	2.5	-2.43	0.015	0.0906
☑	UAGCUUAUCAGACUGAUGUUGAG_hsa-miR-21-5p_1_0_0_0_C/G_59841297_Alt_F_NA	289.1	-6.4644	2.48	-2.6	0.0092	0.0906
☐	AGCUACAUUGUCUGCUGGGU_hsa-miR-221-3p_0_0_0_3_NA_0_NA_F_NA	129.37	-6.0613	2.51	-2.42	0.0156	0.0906

Figure 4.8. Partial table of differentially expressed isomiRs. The grey bar on the right of the image can be used by the user to scroll throughout the complete differential expression table. The current figure only displays the results obtained when the number 21 is browsed.

It is possible to sort the table by any column. It is necessary to select, from this table, the isomiRs desired for functional analysis. For the current dataset, the authors detected the differential expression of some miRNAs, such as miR-34c-5p, miR-126-3p, miR-126-5p, miR-143-3p, miR-379-5p, and miR-1268a). Additionally, it was also mentioned by the authors (and proved in a different study [84]) that, several miRNAs, miR-155-5p, miR-146a-5p and miR-21-5p, target several pathways of the HIV life cycle [83], [84][84]. All the isomiRs originated from these miRNAs were searched in the differential expression. The miR-143-3p, miR-155-5p, miR-146a and miR-21-5p have variants present in the samples that are differentially expressed. Once the complete table and miRNAs counts files are available in the zip file provided by the web application (containing all the results files), miR-34c-5p was searched given the important finding of the published study [20]. Despite being downregulared, it had a low number of counts, which can be due to the version of DESeq used (different from the one used in the study). In Figure 4.8 it is only displayed the selection of the miR-21-5p variants. The total of the isomiRs selected for functional analysis are displayed in Figure 4.9 below.

4.2.4 IsomiRs functional impact

Functional analysis was performed receiving the differentially expressed isomiRs found in annotation analysis (see Figure 4.9).

Selected IsomiRs:

- UGAGAUGAAGCACUGUAGCU_hsa-miR-143-3p_0_1_0_0_NA_0_NA_F_NA
- UGAGAACUGAAUUCCAUGGG_hsa-miR-146a-5p_0_2_0_0_NA_0_NA_F_NA
- UGAGAACUGAAUUCCAUGGGU_hsa-miR-146a-5p_0_1_0_0_NA_0_NA_F_NA
- UAGCUUAUCAGACUGAUGUUG_hsa-miR-21-5p_0_1_0_0_NA_0_NA_F_NA
- UGAGAACUGAAUUCCAUGGGUU_hsa-miR-146a-5p_0_0_0_0_NA_0_NA_F_NA
- UAGCUUAUCAGACUGAUGUU_hsa-miR-21-5p_0_2_0_0_NA_0_NA_F_NA
- AAUGCUAAUCGUGAUAGGGG_hsa-miR-155-5p_0_1_0_2_NA_0_NA_F_NA
- UUAUCAGACUGAUGUUGAC_hsa-miR-21-5p_1_0_0_4_NA_0_NA_F_NA
- UAGCUUAUCAGACUGAUGUUGAG_hsa-miR-21-5p_1_0_0_0_C/G_59841297_Alt_F_NA
- UGAGAUGAAGCACUGUAGCUCU_hsa-miR-143-3p_1_0_0_0_A/T_149429001_Alt_F_NA
- UAGCUUAUCAGACUGAUG_hsa-miR-21-5p_0_4_0_0_NA_0_NA_F_NA
- GAGAACUGAAUUCCAUGGGUU_hsa-miR-146a-5p_0_0_0_1_NA_0_NA_F_NA

Figure 4.9. Selected isomiRs for functional analysis.

The table in Figure 4.10 displays the gene ontologies (GO) represented for the genes from which the isomiRs were originated. It can be observed in the table below that the GO terms found belong to the cellular component domain.

All Fields Search

GO ID	Term	Annotated	Significant	Expected	Rank in classic Fisher	Classic Fisher	Classic KS	Elimination KS
GO:0044464	cell part	12668	9497	9406.27	72	0.0054	<1e-30	<1e-30
GO:0005622	intracellular	11196	8548	8313.28	9	1.0e-11	<1e-30	<1e-30
GO:0043227	membrane-bounded organelle	9607	7397	7133.41	3	4.6e-15	<1e-30	<1e-30
GO:0043229	intracellular organelle	9457	7287	7022.03	2	2.7e-15	<1e-30	<1e-30
GO:0043231	intracellular membrane-bounded organelle	8624	6691	6403.51	1	1.8e-18	<1e-30	<1e-30

Figure 4.10. Gene enrichment table.

4.3 Summary

In this chapter, the benchmarking of the IsomiR Window tool on human datasets was performed. The graphics that allow the quality approval of the samples were displayed. These demonstrate which read lengths are most common in the samples, along with the sncRNAs found and from which harpin arm the miRNAs were originated. These are all interactive, along the user to take a closer look on the objects of each graphic. This quality control of the samples is usually done by the researcher manually, but with the developed tool this information is created automatically.

Additionally, the graphics that expose the miRNA complexity were also displayed. With these, it is possible to know how many reads of each miRNA are present in the samples, what were the main types of isomiRs discovered, what editing events were detected and what were the novel miRNAs, if any was encountered.

The differential expression of the detected isomiRs was also assessed by the IsomiR Window tool, and those found to be differentially expressed proceeded for functional analysis, where the most common GO terms for these isomiRs were displayed.

Chapter 5 – Conclusion

The analysis of small-RNA-seq data has improved over the past years, however the diversity of tools that allow the analysis of this type of data in a user-friendly manner is very reduced, allowing only a limited number of functionalities. Based on the assessment of functionalities provided by available analysis tools (Section 2.2), it was concluded that many of users (researchers) requirements were not provided. These include, ability to process NGS data providing an integrated set of analysis for different data processing stages (e.g. annotation of reads, differential expression, inference regarding biological impact); possibility to define analysis settings and importantly the availability of a user-friendly interface to set-up the analysis. Therefore, this project aimed at filling the gaps identified and was comprised by the design and development of a web application and web services for the IsomiR Window tool, on top of a pipeline with Perl and R scripts (not in the scope of this thesis) that was implemented concurrently. In the following sections the main contributions of this project are described, along with the acquired skills and encountered challenges. As final remarks, some suggestions for future work are given.

5.1 Main contributions

The first contribution was a web application, which allows a comprehensive analysis of sncRNAs with a focus on isomiRs, providing a user-friendly interface with a clean design and logical progress path. This application integrates a pipeline for data analysis and allows the user to perform two modules of analysis, independent from one another, called annotation analysis and functional analysis. Despite independency between the two modules, the application can execute the functional analysis based on the results of the annotation module automatically, if the user so instructs. Additionally, after the analysis, the interface allows to review and save the results that were produced. Throughout the analysis, which can take some time, a progress bar is displayed and updated regularly. Additionally, if an e-mail address is provided, the user receives notifications alerting for the start and end of an analysis, or for errors, providing advice for error solving. This reduces the probability for the user to get frustrated when using the tool.

The interface produces different types of outputs, charts and tables that provide to the user different insights regarding the analyzed data: a) data quality assessment; b) data complexity; c) biological impact. The visualizations provide some interactivity allowing the user, for example, to select important features or to reorder tables.

A virtual machine was also created and contains the web application, along with all dependencies of third-party software required by it. Therefore, all the user needs to do is open the web browser to have access to the functionalities provided by the tool.

The second contribution was the demonstration, through the benchmarking of the tool, that IsomiR Window can be used to assess the quality of datasets, explore the miRNAs and isomiRs found in the datasets, detect the isomiRs that are differentially expressed and analyze their functional impact. The prediction of novel miRNAs is also offered by the tool. This benchmarking was performed on datasets produced in a study [20] that did not explore the isomiRs present in the samples. The IsomiR Window tool identified some of the miRNAs found in the study and also detected a large amount of differentially expressed isomiRs.

It is possible to affirm that this tool, in comparison with other tools currently available, provides several new functionalities that are essential for performing isomiR analysis in an accurate and user-friendly manner.

5.2 Acquired skills

At a personal level, considering that my academic background is in Biology, this project was a big challenge for me. I was first introduced to informatics subjects only two years ago, in the first year of the master degree in Bioinformatics and Computational Biology. The contents of those subjects were introductory and not entirely applicable to this project. It was very gratifying to increase my knowledge in several technologies and to learn concepts that are used in web development. Besides programming the application, I also learned how to create a virtual machine and configure remote access.

5.3 Encountered challenges

A challenge encountered in this work was how the Isomir Window tool could be made available to researchers. Several tools are available in servers with public access, but if the server does not have the disk space to store big amounts of data and processing capacity for multiple users, it results in a service that is hard to use and too slow to provide outputs. Therefore, in this project it was decided that the best option would be to distribute the IsomiR Window tool through a virtual machine. The disadvantage is that the user must import and run the virtual machine in his computer. However, all dependencies are already resolved and the user only needs to start the virtual machine and access IsomiR Window web application through the available browser to start an analysis.

Learning Laravel, having no knowledge about this framework, or any other frameworks also presented to be a challenge. Certain functionalities provided by the framework were challenging to program, such as finding out how the commands available from the terminal could be invoked. Additionally, how to connect both analyses (annotation and functional) also presented to be a difficulty. 1.1. Another challenge was hard-disk space available in the virtual machine. For the good functioning of the machine, 150gb should be available as hard-disk space. However, because the image of the machine is to be downloaded, this amount of space would be impractical. Therefore, a considerable amount of hard-disk space was attributed to the machine, allowing the user to increase this value if necessary. The instructions to achieve this are given in Appendix A, Section 4.

5.4 Future work

For the application to become more complete and provide a better experience to the user, there are some aspects that can be improved.

The tool should allow the analysis of the datasets coming from more organisms than just the one provided by the tool. The current tool is innovative since, in contrast to other isomiR analysis tools, allows analyzing in parallel several datasets for two experimental conditions. Nevertheless, allowing the user to set the desired number of conditions in the design, allowing the parallel analysis of all studied conditions should be considered in the near future.

The functionalities associated with focus area number 4 of the UED (in Figure 3.2), that would provide more details about the isomiRs found in the samples, should also be considered as future work. These details could also be demonstrated through visualizations produced by SVG.

New graphics that inform the user about the total number of reads in the samples, the total number of mapped reads in the genome and the number of total reads annotated could also be created.

References

[1] J. Zhang, R. Chiodini, A. Badr, and G. Zhang, "The impact of next-generation sequencing on genomics," *Journal of Genetics and Genomics*, vol. 38, no. 3. pp. 95–109, 2011.

[2] V. N. Kim, "Small RNAs: classification, biogenesis, and function.," *Molecules and cells*, vol. 19, no. 1, pp. 1–15, 2005.

[3] R. D. Morin, M. D. O'Connor, M. Griffith, F. Kuchenbauer, A. Delaney, A. L. Prabhu, Y. Zhao, H. McDonald, T. Zeng, M. Hirst, C. J. Eaves, and M. A. Marra, "Application of massively parallel sequencing to microRNA profiling and discovery in human embryonic stem cells," *Genome Research*, vol. 18, no. 4, pp. 610–621, 2008.

[4] C. T. Neilsen, G. J. Goodall, and C. P. Bracken, "IsomiRs - The overlooked repertoire in the dynamic microRNAome," *Trends in Genetics*, vol. 28, no. 11. pp. 544–549, 2012.

[5] S. L. Ameres and P. D. Zamore, "Diversifying microRNA sequence and function.," *Nature reviews. Molecular cell biology*, vol. 14, no. 8, pp. 475–88, 2013.

[6] M. Karali, M. Persico, M. Mutarelli, A. Carissimo, M. Pizzo, V. Singh Marwah, C. Ambrosio, M. Pinelli, D. Carrella, S. Ferrari, D. Ponzin, V. Nigro, D. Di Bernardo, and S. Banfi, "High-resolution analysis of the human retina miRNome reveals isomiR variations and novel microRNAs," *Nucleic Acids Research*, vol. 44, no. 4, pp. 1525–1540, 2016.

[7] M. Hackenberg, M. Sturm, D. Langenberger, J. M. Falcón-Pérez, and A. M. Aransay, "miRanalyzer: A microRNA detection and analysis tool for next-generation sequencing experiments," *Nucleic Acids Research*, vol. 37, no. SUPPL. 2, pp. 68–76, 2009.

[8] Y. Zhang, B. Xu, Y. Yang, R. Ban, H. Zhang, X. Jiang, H. J. Cooke, Y. Xue, and Q. Shi, "Cpss: A computational platform for the analysis of small rna deep sequencing data," *Bioinformatics*, vol. 28, no. 14, pp. 1925–1927, 2012.

[9] S. Cho, I. Jang, Y. Jun, S. Yoon, M. Ko, Y. Kwon, I. Choi, H. Chang, D. Ryu, B. Lee, V. N. Kim, W. Kim, and S. Lee, "MiRGator v3.0: A microRNA portal for deep sequencing, expression profiling and mRNA targeting," *Nucleic Acids Research*, vol. 41, no. D1, pp. 252–257, 2013.

[10] W. C. Cheng, I. F. Chung, T. S. Huang, S. T. Chang, H. J. Sun, C. F. Tsai, M. L. Liang, T. T. Wong, and H. W. Wang, "YM500: A small RNA sequencing (smRNA-seq) database for microRNA research," *Nucleic Acids Research*, vol. 41, no. D1, pp. 285–294, 2013.

[11] L. Pantano, X. Estivill, and E. Martí, "SeqBuster, a bioinformatic tool for the processing and analysis of small RNAs datasets, reveals ubiquitous miRNA modifications in human embryonic cells," *Nucleic Acids Research*, vol. 38, no. 5, 2009.

[12] D. T. Humphreys and C. M. Suter, "MiRspring: A compact standalone research tool for analyzing miRNA-seq data," *Nucleic Acids Research*, vol. 41, no. 15, 2013.

[13] L. F. V. De Oliveira, A. P. Christoff, and R. Margis, "isomiRID: A framework to identify microRNA isoforms," *Bioinformatics*, vol. 29, no. 20, pp. 2521–2523, 2013.

[14] H. Muller, M. J. Marzi, and F. Nicassio, "IsomiRage: from functional classification to differential expression of miRNA isoforms," *Frontiers in bioengineering and biotechnology*, vol. 2, no. 38, pp. 1-10, 2014.

[15] A. Kaushik, S. Saraf, S. K. Mukherjee, and D. Gupta, "miRMOD: a tool for identification and analysis of 5' and 3' miRNA modifications in Next Generation Sequencing small RNA data," *PeerJ*, vol. 3, no. 1332, pp. 1-15, 2015.

[16] L. Guo, J. Yu, T. Liang, and Q. Zou, "miR-isomiRExp: a web-server for the analysis of expression of miRNA at the miRNA/isomiR levels," *Scientific Reports*, vol. 6, no. 23700, pp. 1-7, 2016.

[17] G. Urgese, G. Paciello, A. Acquaviva, and E. Ficarra, "isomiR-SEA: an RNA-Seq analysis tool for miRNAs/isomiRs expression level profiling and miRNA-mRNA interaction sites evaluation," *BMC Bioinformatics*, vol. 17, no. 148, pp. 1-13, 2016.

[18] Y. Zhang, Q. Zang, H. Zhang, R. Ban, Y. Yang, F. Iqbal, A. Li, and Q. Shi, "DeAnnIso: a tool for online detection and annotation of isomiRs from small RNA sequencing data," *Nucleic Acids Research*, vol. 44, no. 427 pp. 166-175, 2016.

[19] G. Sablok, I. Milev, G. Minkov, I. Minkov, C. Varotto, G. Yahubyan, and V. Baev, "IsomiRex: Web-based identification of microRNAs, isomiR variations and differential expression using next-generation sequencing datasets," *FEBS Letters*, vol. 587, no. 16, pp. 2629–2634, 2013.

[20] A. J. Amaral, J. Andrade, R. B. Foxall, P. Matoso, A. M. Matos, R. S. Soares, C. Rocha, C. G. Ramos, R. Tendeiro, A. Serra-Caetano, J. A. Guerra-Assunção, M. Santa-Marta, J. Gonçalves, M. Gama-Carvalho, and A. E. Sousa, "miRNA profiling of human naive CD4 T cells links miR-34c-5p to cell activation and HIV replication," *The EMBO Journal*, vol. 43, no. 10, pp. 1–15, 2016.

[21] H. Beyer and K. Holtzblatt, *Contextual design: Defining customer-centered systems.* Concord: Morgan Kaufmann, 1998, pp. 318-345.

[22] E. Wienholds and R. H. A. Plasterk, "MicroRNA function in animal development," *FEBS Letters*, vol. 579, no. 26. pp. 5911–5922, 2005.

[23] Y. Lee, M. Kim, J. Han, K.-H. Yeom, S. Lee, S. H. Baek, and V. N. Kim, "MicroRNA genes are transcribed by RNA polymerase II," *The EMBO Journal*, vol. 23, no. 20, pp. 4051–4060, 2004.

[24] X. Cai, C. H. Hagedorn, and B. R. Cullen, "Human microRNAs are processed from capped, polyadenylated transcripts that can also function as mRNAs," *RNA* , vol. 10, no. 12, pp. 1957–1966, 2004.

[25] Z. Xie, E. Allen, N. Fahlgren, A. Calamar, S. a Givan, and J. C. Carrington, "Expression of Arabidopsis MIRNA genes.," *Plant physiology*, vol. 138, no. 4, pp. 2145–2154, 2005.

[26] R. I. Gregory, K.-P. Yan, G. Amuthan, T. Chendrimada, B. Doratotaj, N. Cooch, and R. Shiekhattar, "The Microprocessor complex mediates the genesis of microRNAs," *Nature*, vol. 432, no. 7014, pp. 235–240, 2004.

[27] A. Grishok, A. E. Pasquinelli, D. Conte, N. Li, S. Parrish, I. Ha, D. L. Baillie, A. Fire, G. Ruvkun, and C. C. Mello, "Genes and mechanisms related to RNA interference regulate expression of the small temporal RNAs that control C. elegans developmental timing," *Cell*, vol. 106, no. 1, pp. 23–34, 2001.

[28] S. W. Eichhorn, H. Guo, S. E. McGeary, R. A. Rodriguez-Mias, C. Shin, D. Baek, S. hao Hsu, K. Ghoshal, J. Villén, and D. P. Bartel, "MRNA Destabilization Is the dominant effect of mammalian microRNAs by the time substantial repression ensues," *Molecular Cell*, vol. 56, no. 1, pp. 104–115, 2014.

[29] D. D. Jima, J. Zhang, C. Jacobs, K. L. Richards, C. H. Dunphy, W. W. L. Choi, W. Y. Au, G. Srivastava, M. B. Czader, D. A. Rizzieri, A. S. Lagoo, P. L. Lugar, K. P. Mann, C. R. Flowers, L. Bernal-Mizrachi, K. N. Naresh, A. M. Evens, L. I. Gordon, M. Luftig, D. R. Friedman, J. B. Weinberg, M. A. Thompson, J. I. Gill, Q. Liu, T. How, V. Grubor, Y. Gao, A. Patel, H. Wu, J. Zhu, G. C. Blobe, P. E. Lipsky, A. Chadburn, and S. S. Dave, "Deep sequencing of the small RNA transcriptome of normal and malignant human B cells identifies hundreds of novel microRNAs," *Blood*, vol. 116, no. 23, pp. 118-127, 2010.

[30] D. H. Jeong and P. J. Green, "Methods for validation of miRNA sequence variants and the cleavage of their targets," *Methods*, vol. 58, no. 2, pp. 135–143, 2012.

[31] G. C. Tan and N. Dibb, "IsomiRs have functional importance," *Malaysian Journal of Pathology*, vol. 37, no. 2, pp. 73–81, 2015.

[32] F. Kuchenbauer, R. D. Morin, B. Argiropoulos, O. I. Petriv, M. Griffith, M. Heuser, E. Yung, J. Piper, A. Delaney, A. L. Prabhu, Y. Zhao, H. McDonald, T. Zeng, M. Hirst, C. L. Hansen, M. A. Marra, and R. K. Humphries, "In-depth characterization of the microRNA transcriptome in a leukemia progression model," *Genome Research*, vol. 18, no. 11, pp. 1787–1797, 2008.

[33] G. Martin and W. Keller, "RNA-specific ribonucleotidyl transferases," *RNA* , vol. 13, no. 11, pp. 1834–1849, 2007.

[34] F. O. Kelly, L. Marignol, A. Meunier, T. H. Lynch, and A. S. Perry, "MicroRNAs as putative

mediators of treatment response in prostate cancer," *Nature Reviews Urology*, vol. 9, no. 7, pp. 397–407, 2012.

[35] S. K. Wyman, E. C. Knouf, R. K. Parkin, B. R. Fritz, D. W. Lin, L. M. Dennis, M. A. Krouse, P. J. Webster, and M. Tewari, "Post-transcriptional generation of miRNA variants by multiple nucleotidyl transferases contributes to miRNA transcriptome complexity," *Genome Research*, vol. 21, no. 9, pp. 1450–1461, 2011.

[36] B. W. Han, J. H. Hung, Z. Weng, P. D. Zamore, and S. L. Ameres, "The 3′-to-5′ exoribonuclease nibbler shapes the 3′ ends of microRNAs bound to drosophila argonaute1," *Current Biology*, vol. 21, no. 22, pp. 1878–1887, 2011.

[37] K. Nishikura, "Functions and regulation of RNA editing by ADAR deaminases," *Annual review of biochemistry*, vol. 79, pp. 321–349, 2010.

[38] B. M. Ryan, A. I. Robles, and C. C. Harris, "Genetic variation in microRNA networks: the implications for cancer research," *Nature reviews. Cancer*, vol. 10, no. 6, pp. 389–402, 2010.

[39] Y. K. Kim, I. Heo, and V. N. Kim, "Modifications of Small RNAs and their associated proteins," *Cell*, vol. 143, no. 5. pp. 703–709, 2010.

[40] D. P. Bartel, "MicroRNA target recognition and regulatory functions," *Cell*, vol. 136, no. 2, pp. 215–233, 2009.

[41] H. Seitz, M. Ghildiyal, and P. D. Zamore, "Argonaute loading improves the 5′ precision of both microRNAs and their miRNA strands in flies," *Current Biology*, vol. 18, no. 2, pp. 147–151, 2008.

[42] M. Ghildiyal, J. Xu, H. Seitz, Z. Weng, and P. D. Zamore, "Sorting of drosophila small silencing RNAs partitions microRNA* strands into the RNA interference pathway," *RNA*, vol. 16, no. 1, pp. 43–56, 2010.

[43] N. Cloonan, S. Wani, Q. Xu, J. Gu, K. Lea, S. Heater, C. Barbacioru, A. L. Steptoe, H. C. Martin, E. Nourbakhsh, K. Krishnan, B. Gardiner, X. Wang, K. Nones, J. A. Steen, N. A. Matigian, D. L. Wood, K. S. Kassahn, N. Waddell, J. Shepherd, C. Lee, J. Ichikawa, K. McKernan, K. Bramlett, S. Kuersten, and S. M. Grimmond, "MicroRNAs and their isomiRs function cooperatively to target common biological pathways," *Genome biology*, vol. 12, no. 12, pp. 1-20, 2011.

[44] S. L. Fernandez-Valverde, R. J. Taft, and J. S. Mattick, "Dynamic isomiR regulation in Drosophila development," *RNA*, vol. 16, no. 10, pp. 1881–1888, 2010.

[45] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, G. P. Data, T. Sam, and 1000 Genome Project Data Processing Subgroup, "The sequence alignment / map format and SAMtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.

[46] A. Kozomara and S. Griffiths-Jones, "MiRBase: Annotating high confidence microRNAs using deep sequencing data," *Nucleic Acids Research*, vol. 42, no. D1, pp. 68–73, 2014.

[47] The RNAcentral Consortium, "RNAcentral: a comprehensive database of non-coding RNA sequences," *Nucleic Acids Research*,vol. 45, no. 45, pp. 128-134, 2016.

[48] M. R. Friedländer, S. D. MacKowiak, N. Li, W. Chen, and N. Rajewsky, "MiRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades," *Nucleic Acids Research*, vol. 40, no. 1, pp. 37–52, 2012.

[49] S. H. Bernhart, I. L. Hofacker, and P. F. Stadler, "Local RNA base pairing probabilities in large sequences," *Bioinformatics*, vol. 22, no. 5, pp. 614–615, 2006.

[50] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: A Bioconductor package for differential expression analysis of digital gene expression data," *Bioinformatics*, vol. 26, no. 1, pp. 139–140, 2009.

[51] S. Anders, W. Huber, A. S, and H. W, "DESeq: Differential expression analysis for sequence count data," *Genome biology*, vol. 11, no. 10, pp. 1-12, 2010.

[52] T. J. Hardcastle and K. A. Kelly, "baySeq: Empirical Bayesian methods for identifying differential expression in sequence count data," *BMC Bioinformatics*, vol. 11, no. 422, pp. 1-14, 2010.

[53] N. Leng, J. A. Dawson, J. A. Thomson, V. Ruotti, A. I. Rissman, B. M. G. Smits, J. D. Haag, M. N. Gould, R. M. Stewart, and C. Kendziorski, "EBSeq: An empirical Bayes hierarchical model for inference in RNA-seq experiments," *Bioinformatics*, vol. 29, no. 8, pp. 1035–1043, 2013.

[54] J. Krüger and M. Rehmsmeier, "RNAhybrid: MicroRNA target prediction easy, fast and flexible," *Nucleic Acids Research*, vol. 34, no. 34, pp. 451-454, 2006.

[55] D. Betel, M. Wilson, A. Gabow, D. S. Marks, and C. Sander, "The microRNA.org resource: Targets and expression," *Nucleic Acids Research*, vol. 36, no. 36, pp. 149-153, 2008.

[56] V. Agarwal, G. W. Bell, J. W. Nam, and D. P. Bartel, "Predicting effective microRNA target sites in mammalian mRNAs," *eLife*, vol. 4, no. 4, pp. 1–38, 2015.

[57] M. Kertesz, N. Iovino, U. Unnerstall, U. Gaul, and E. Segal, "The role of site accessibility in microRNA target recognition," *Nature Genetics*, vol. 39, no. 10, pp. 1278–1284, 2007.

[58] D. Huang, B. T. Sherman, Q. Tan, J. R. Collins, W. G. Alvord, J. Roayaei, R. Stephens, M. W. Baseler, H. C. Lane, and R. A. Lempicki, "The DAVID gene functional classification tool: a novel biological module-centric algorithm to functionally analyze large gene lists," *Genome Biology*, vol. 8, no. 9, p. 1-16, 2007.

[59] Bioconductor, "The open source software for bioinformatics", 2003. [Online]. Available: https://www.bioconductor.org/. [Accessed: 12 Aug-2017].

[60] R Core Team, "R: A Language and Environment for Statistical Computing", 2014. [Online]. Available: http://www.R-project.org/. [Accessed: 12-Aug-2017].

[61] A. Alexa and J. Rahnenführer, "Gene set enrichment analysis with topGO," R package version 2.28.0.

[62] S. Falcon and R. Gentleman, "Using GOstats to test gene lists for GO term association," *Bioinformatics*, vol. 23, no. 2, pp. 257–258, 2007.

[63] J. Nielsen, *Usability Engineering*, San Francisco: Morgan Kaufmann, 1993, pp. 115-163.

[64] R. Nixon, *Learning PHP, MySQL, JavaScript, and CSS*, Sebastopol: O'Reilly, 2012, pp. 1-41.

[65] W3 Schools, "HTML", 2017. [Online]. Available: https://www.w3schools.com/html/. [Accessed: 01-Aug-2017].

[66] W3 Schools, "CSS", 2017. [Online]. Available: https://www.w3schools.com/css/. [Accessed: 01-Aug-2017].

[67] D. Flanagan, *JavaScript: The Definitive Guide 6th Edition*, Sebastopol: O'Reilly, 2011, pp. 1-70.

[68] A. T. Holdener, *Ajax: The Definitive Guide*, Sebastopol: O'Reilly, 2008, pp. 3-36.

[69] W3 Consortium, "Extensible Markup Language (XML)", 2016. [Online]. Available: https://www.w3.org/XML/. [Accessed: 01-Aug-2017].

[70] W3 Consortium, "Web Services Architecture", 2014. [Online]. Available: https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest. [Accessed: 01-Aug-2017].

[71] W3 Consortium, "Web Services Description Language (WSDL)", 2003. [Online]. Available: https://www.w3.org/TR/2003/WD-wsdl20-20031110/. [Accessed: 01-Aug-2017].

[72] L. Richardson and S. Ruby, *RESTful Web Services*, Sebastopol: O'Reilly, 2007, pp. 5-80.

[73] B. D. Sklar and A. Trachtenberg, *PHP Cookbook*, Sebastopol: O'Reilly, 2002, pp. 1-30.

[74] Laravel , 2017. [Online]. Available: https://laravel.com/docs/5.5. [Accessed: 05-Aug-2017].

[75] Symfony, 2017. [Online]. Available: http://symfony.com/. [Accessed: 05-Aug-2017].

[76] CodeIgniter, 2017. [Online]. Available: https://codeigniter.com/. [Accessed: 05-Aug-2017].

[77] Internet Alchemy, "What Are The Benefits of MVC?," 2008. [Online]. Available: http://blog.iandavis.com/2008/12/what-are-the-benefits-of-mvc/. [Accessed: 05-Aug-2017].

[78] "Programming Using the MVC Architecture", 2015. [Online]. Available: http://www.c-sharpcorner.com/UploadFile/201fc1/programming-in-java-using-the-mvc-architecture/. [Accessed: 11-Sep-2017].

[79] PHP-FIG, "PHP Framework Interop Group", 2016. [Online]. Available: http://www.php-

fig.org/. [Accessed: 07-Sep-2017].

[80] Stack Overflow, 2017. [Online]. Available: https://pt.stackoverflow.com/. [Accessed: 07-Sep-2017].

[81] XAMPP, 2017. [Online]. Available: https://www.apachefriends.org/pt_br/index.html. [Accessed: 08-Aug-2017].

[82] I. Viegas, "Identifying the sequence complexity of miRNAs and their functional impact in small-RNA-seq data", Master thesis, not yet evaluated.

[83] A. Eulalio, E. Huntzinger, and E. Izaurralde, "Getting to the root of miRNA-mediated gene silencing," *Cell*, vol. 132, no. 1. pp. 9–14, 2008.

[84] S. Barichievy, J. Naidoo, and M. M. Mhlanga, "Non-coding RNAs and HIV: Viral manipulation of host dark matter to shape the cellular environment," *Frontiers in Genetics*, vol. 6, no. 108, pp. 1-11 , 2015.

Appendix A – User guide for the IsomiR Window tool

This section describes the needed guidelines for the correct use of IsomiR Window tool. The guidelines will be divided into two different parts. First, the guidelines for the installation of the IsomiR Window virtual machine (VM) in the guest machine will be provided. After this, the guidelines for the usage of the Web application will be given. The last is similar to the guidelines provided in the web application itself.

1. Installation of VirtualBox software

1.1. Go to https://www.virtualbox.org/ website and proceed to the Downloads section.
1.2. Download the VirtualBox Platform Package that is suitable for your operating system and follow the instructions for installation.
1.3. Make sure to enable Intel Virtualization Technology (VT-x) and AMD-V in BIOS of your computer.

2. Import the VM to VirtualBox software

2.1. Go to the https://isomir.fc.ul.pt/ website and download the files available.
2.2. After uncompressing the downloaded file open the VirtualBox software.
2.3. In VirtualBox, open the File tab and click on Import Appliance.
2.4. Select the VM file and make sure that is called IsomirWindow.ova. Click Import.
2.5. Once the VM has finished loading to VirtualBox, just click Start. Keep in mind that the username and the password are both called *user*.

3. Start IsomiR Window

3.1. After booting the VM, open a Terminal and type the following command:

```
php /opt/lampp/htdocs/isomirwindow/artisan serve
```

3.2. Now the IsomiR Window can be accessed by typing http://127.0.0.1:8000 into the browser.

4. Increase the VM hard-disk space (optional)

4.1. In the Host machine's Terminal insert the following commands:

```
cd /path/to/vbox/disks
VBoxManage modifyhd IsomirWindow.vdi --resize <desired total size>
```

The <desired total size> is in megabytes.

4.2. Start the Isomir Window VM.
4.3. Open a Terminal and type the following command:

```
sudo gparted
```

4.4. In the GParted interface right-click on the partition called /dev/sda1 and click Resize/Move.
4.5. Fill in the size as required. Click Resize and then click Apply all operations
4.6. After rebooting the VM, the hard-disk space is now increased.

5. Create a shared folder between the host and virtual machine (optional)

5.1. Add your user to the vboxusers group in the host machine. To do this, open a Terminal in the Host machine and insert the following commands:

```
sudo usermod –a –G vboxusers <userhostname>
groupadd vboxsf
sudo usermod –a –G vboxsf <userhostname>
```

5.2. Create the folder, on Host, to be shared with the VM. To do this, open a Terminal in the Host machine and insert the following commands:

```
mkdir /path-to-shared-folder/shared-folder
sudo chmod -R 777 /path-to-shared-folder/shared-folder
sudo chown -R userhostname:userhostname /path-to-shared-folder/shared-folder
```

5.3. Inside the VM, add shared folder by clicking in the folder icon on the bottom right of the window and select Shared Folder Settings…

5.4. In the panel, click in Add new shared folder.

5.5. On the folder path field, point to the folder created in section 1.4.2. Select Auto-mount and Make Permanent.

5.6. In the VM, open a Terminal and insert the following commands:

```
mkdir /media/sf_VMFiles
sudo mount -t vboxsf <shared-folder> /media/sf_VMFiles
sudo usermod -G vboxsf -a user
```

5.7. Restart the VM and an ISO image of the sf_VMFiles shared folder is mounted.

Appendix B – IsomiR Window virtual machine creation

IsomirR Window tool is accessible through a web browser inside a virtual machine (VM). In this appendix, all the steps taken in the creation of the virtual machine are explained.

It is important to note that, for the correct functioning of the VM, it is necessary to download the compiled file, available in http://isomir.fc.ul.pt . The compiled file contains a folder corresponding to the application itself and additional files, which are mentioned and required in some of the steps in the IsomiR Window VM creation described below.

The creation of the VM consists in 8 steps. The first step explains the installation of VirtualBox software, that supports the execution of the VM. The second step is the creation of the IsomiR Window VM, based in the Ubuntu Linux distribution, in the VirtualBox software. The third step is the installation of Ubuntu software in the IsomiR Window VM. The fourth and fifth steps consist in the installation of VirtualBox software packages, namely Extension pack and Guest Additions pack, to add functionalities that make the VM more useful and easy to use. The sixth step consists in the installation of all the third-party software packages required for the IsomiR Window tool to work. Every step for the installation of each software is given. The seventh step is the update of the IsomiR Window tool and consists in transferring the application folder to the VM. The eighth and final step is the export of the VM, which creates an OVF file, which can be loaded and executed in the user's computer using VirtualBox 5.1. Each one of the eighth steps will be explained in detail below.

2. Installation of VirtualBox software

2.1. Go to https://www.virtualbox.org/ Website and proceed to the Downloads section.
2.2. Download the VirtualBox Platform Package that is suitable for your operating system and follow the instructions for installation.
2.3. Make sure to enable Intel Virtualization Technology (VT-x) and AMD-V in BIOS of your computer.

3. Creation of IsomiR Window VM

3.1. Go to https://www.ubuntu.com/download/desktop and download Ubuntu 16.04.2 LTS ISO image.
3.2. Open VirtualBox and click on New to start a new VM.
3.3. In the Name field insert Isomir Window. Type must be set to Linux and Version to Ubuntu. Click Next.
3.4. The amount of memory RAM allocated will vary according to your computer, but the recommended is 8GB. Click Next.
3.5. Select Create a virtual hard drive now. Click Create.
3.6. Select VDI (VirtualBox Disk Image). Click Next.
3.7. Select Dynamically allocated. Click Next.
3.8. The selection of the size of the virtual hard drive is also dependent on your computer. The minimum recommend size is 60GB, but if possible and needed, feel free to increase this value. Click Create.
3.9. Right click on your VM (Isomir Window) and choose Settings.
3.10. Go to Storage and click on the field that has a CD icon and says Empty.

3.11. In the right side of the panel (Attributes), click in the icon with a folder icon and select the ISO image downloaded in the first step of this section.
3.12. Still on Settings, click on System.
3.13. Go to the Processor tab and attribute to your VM at least 2 processors (CPUs).
3.14. Go to Acceleration tab and make sure that the options Enable VT-x/AMD-v and Enable Nested Paging are selected.
3.15. Click Ok to exit Settings.

4. Installation of IsomiR Window VM

4.1. In the VirtualBox panel, select your VM and click on Start.
4.2. Select English as the language and click on Install Ubuntu.
4.3. Select Download updates while installing Ubuntu and click Continue.
4.4. In Installation Type, choose the Erase disk and install Ubuntu option. Click Install Now.
4.5. Pick your time zone and click Continue.
4.6. Pick your keyboard language and click Continue.
4.7. Configure the names and passwords to be used in the VM as listed below and click Continue.

Your name : **user**
Your computer's name: **IsomirWindow**
Pick a username: **user**
Password: **user**

4.8. After the installation is complete click on Restart Now.
4.9. After restart, power off the VM in the Virtuabox panel.

5. Installation of VirtualBox Extension Pack

5.1. Go to https://www.virtualbox.org/ Website and proceed to the Downloads section.
5.2. Search for the VirtualBox 5.1.26 Oracle VM VirtualBox Extension Pack and download it.
5.3. To install the extension, simply double-click on the package file, which as a *.vbox-extpack* extension, and a Network Operations Manager window will appear, guiding you through the required steps.

6. Installation of VirtualBox Guest Additions and creation of a shared folder

6.1. In the VirtualBox panel, select your VM and click on Start.
6.2. Type the following command in the Terminal:

```
sudo apt install dkms
```

6.3. Click on the tab called Devices, which is located in the left upper corner of the VM window and choose Insert Guest Additions CD image.
6.4. Click run and enter the password (user).
6.5. Restart the VM.
6.6. Click eject in the Guest additions ISO image.

6.7. In the bottom right corner of the window, right-click in the folder icon and select Shared Folders Settings.

6.8. Add a new folder and define the path to the folder you wish to share with the VM, naming it VMfiles.

6.9. Type the following commands in the Terminal:

```
cd Desktop
mkdir shared
sudo mount –t vboxsf VMfiles shared
```

6.10. Go to the Virtualbox panel and right-click on your VM.

6.11. Choose Settings, and in the General section and go the Advanced tab.

6.12. Define the fields Shared Clipboard and Drag'n'Drop as bidirectional.

7. Installation of third-party software packages required for the IsomiR Window tool

7.1. XAMPP

XAMPP is a PHP development environment. It is free and easy to install the Apache distribution containing MariaDB, PHP, and Perl.

7.1.1. Go to https://www.apachefriends.org/download.html and download XAMPP for Linux 7.1.8.

7.1.2. Type the following commands in the Terminal:

```
cd Downloads
sudo chmod +x xampp-linux-x64-7.1.8-0-installer.run
sudo ./xampp-linux-x64-7.1.8-0-installer.run
```

7.1.3. Maintain the default values that are in the XAMPP installation window that appears. Keep clicking on next to complete the installation.

7.1.4. To verify if XAMPP was correctly installed, type the following commands in the Terminal:

```
sudo /opt/lampp/lampp start
```

7.1.5. Make XAMPP start on machine boot by typing the following command in the Terminal:

```
sudo nano /etc/rc.local
```

And enter the command `sudo /opt/lampp/lampp start` before the line `exit 0`.

7.1.6. Add PHP (automatically installed by XAMPP) to the environment variable PATH by inserting the following commands in the Terminal:

```
nano ~/.bash_profile
export PATH=$PATH:/opt/lampp/bin
source ~/.bash_profile
```

7.2. Composer

Composer is a tool for dependency management in PHP. It allows the declaration of the libraries that the project depends on and it manages (install/update) them.

7.2.1. Go the https://getcomposer.org/doc/00-intro.md#installation-linux-unix-osx Website and consult the documentation provided for the installation of Composer globally.

7.2.2. If you wish, just follow the commands below:

```
php -r "copy('https://getcomposer.org/installer', 'composer- setup.php');"
php -r "if (hash_file('SHA384', 'composer-setup.php')
===='669656bab3166a7aff8a7506b8cb2d1c292f042046c5a994c43155c0be6190fa0355160742ab2e
1c88d40d5be660b410') { echo 'Installer verified'; } else { echo 'Installer corrupt';
unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
sudo mv composer.phar /usr/local/bin/composer
```

7.3. Laravel 5.4

Laravel is a PHP framework that utilizes Composer to manage its dependencies.

7.3.1. Go the https://laravel.com/docs/5.4 Website and consult the documentation provided for the installation.

7.3.2. Open a Terminal and type the following commands for installation:

```
composer global require "laravel/installer"
nano ~/.bash_profile
export PATH="$PATH:/opt/lampp/bin:~/.config/composer/vendor/bin"
source ~/.bash_profile
```

7.3.3. Change the permissions of the folder were the Laravel project will be created. Create the project.

```
sudo chown user -R /opt/lampp/htdocs
cd /opt/lampp/htdocs
```

7.4. Bowtie 1.2.1.1

Bowtie is a fast, memory-efficient short read aligner. It aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small.

7.4.1. Go to http://bowtie-bio.sourceforge.net/index.shtml, download the zip file named bowtie-1.2.1.1-linux-x86_64.zip and extract it.

7.4.2. In the Terminal type the following command:

```
sudo apt install bowtie
```

7.4.3. Go to the bowtie-1.2.1.1 folder and delete all the files present in the indexes subfolder, once they were only example files not important to our tool.

7.4.4. Go to the isomirWindow.zip (available in http://isomir.fc.ul.pt), extract it and move the GRCh38.dna.fa to **home/user/Downloads/bowtie-1.2.1.1/genomes**.

7.4.5. In the Terminal type the following command:

```
cd ~/Downloads/bowtie-1.2.1.1/genomes
bowtie-build -f GRCh38.dna.fa GRCh38
```

7.5. HT_Seq

HTSeq is a Python package that provides infrastructure to process data from high-throughput sequencing assays.

7.5.1. http://www-huber.embl.de/HTSeq/doc/install.html

7.5.2. In the Terminal type the following commands:

```
sudo apt-get install python-numpy
sudo apt-get install python-matplotlib
sudo apt-get install python-htseq
```

7.6. SAM Tools and BCFtools

SAM Tools provide various utilities for manipulating alignments in the SAM format, including sorting, merging, indexing and generating alignments in a per-position format.

BCFtools is a set of utilities that manipulate variant calls in the Variant Call Format (VCF) and its binary counterpart BCF.

7.6.1. Go http://www.htslib.org/download/, download samtools-1.5 tar and bcftools-1.5 tar and extract it.

7.6.2. In the Terminal type the following commands:

```
sudo apt-get install libncurses-dev
sudo apt-get install zlib1g-dev
sudo apt-get install libbz2-dev
sudo apt-get install liblzma-dev
sudo apt-get install libxml2-dev
sudo apt-get install libcurl4-openssl-dev
cd ~/Downloads/samtools-1.5
./configure --prefix=/home/user
make
make install
cd ~/Downloads/bcftools-1.5
./configure --prefix=/home/user
make
make install
```

7.7. R

R is an open source programming language and software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing.

7.7.1. In the Terminal type the following commands:

```
sudo apt install r-base-core
sudo R
source("https://bioconductor.org/biocLite.R")
biocLite("DESeq2")
biocLite("topGO")
biocLite("biomaRt")
biocLite("org.Hs.eg.db")
install.packages("pheatmap")
install.packages("locfit")
install.packages("gplots")
```

7.8. MirDeep2

MiRDeep2 is a tool which discovers miRNA genes by analyzing sequenced RNAs. The tool reports known and hundreds of novel microRNAs with high accuracy in seven species representing the major animal clades.

7.8.1. In the Terminal insert the following commands:

```
wget https://repo.continuum.io/archive/Anaconda2-4.4.0-Linux-x86_64.sh
chmod +x Anaconda2-4.4.0-Linux-x86_64.sh
./Anaconda2-4.4.0-Linux-x86_64.sh
# Choose default options

cd /home/user/anaconda2/bin
./conda install -c bioconda mirdeep2
nano ~/.bashrc
export PATH=$PATH:~/anaconda2/bin
```

7.8.2. Open the isomirWindow.zip (available in http://isomir.fc.ul.pt) and extract all the files inside the MirdeepSoftware folder to the directory **/home/user/anaconda2/bin**, replacing of the existing files.

7.9. MiRanda

MiRanda algorithm is based on a comparison of miRNAs complementarity to 3'UTR regions. The binding energy of the duplex structure, evolutionary conservation of the whole target site and its position within 3”UTR are calculated and account for a final result which is a weighted sum of match and mismatch scores for base pairs and gap penalties.

7.9.1. Go to the http://www.microrna.org/microrna/getDownloads.do and proceed to the section called miRanda Downloads.

7.9.2. Download the file called miRanda-aug2010 and extract it.

7.9.3. In the Terminal type the following commands:

```
cd ~/Downloads/miRanda-3.3a
./configure
make
sudo make install
```

7.10. Bioperl

The Bioperl Project is an international association of users & developers of open source Perl tools for bioinformatics, genomics and life science.

7.10.1. In the Terminal type the following commands:

```
cd
sudo cpan
install Number::Range
install Bio::Seq
```

7.11. GATK

GATK is a toolkit that offers a wide variety of tools with a primary focus on variant discovery and genotyping. Its powerful processing engine and high-performance computing features make it capable of taking on projects of any size.

7.11.1. Go to https://software.broadinstitute.org/gatk/download/ and download the file called GATK 3.8-0.
7.11.2. Extract the file, making sure that the file named GenomeAnalysisTK.jar is in the Downloads folder.
7.11.3. In the Terminal type the following command:

```
sudo apt install openjdk-8-jre-headless
```

8. Update of the IsomiR Window tool

8.1. Go to the isomirWindow.zip (available in http://isomir.fc.ul.pt) and copy the folder called isomirwindow to the following path**: /opt/lamp/htdocs**.
8.2. In the Terminal type the following commands:

```
cd /opt/lampp/htdocs/isomirwindow
composer update
sudo /opt/lamp/lamp start
```

8.3. Open the file /**opt/lampp/htdocs/isomirwindow/vendor/symphony/process/Process.php** in a text editor, go to line 148 and replace the **$timeout** variable value with *null*.

8.4. Go to localhost/phpmyadmin and create a new database with the following settings:

```
Database name: isomirWindow_DB
Collation: utf8_unicode_ci
```

8.5. In the Terminal insert the following commands:

```
cd /opt/lampp/htdocs/isomirwindow
php artisan migrate
```

8.6. Update several values in the php.ini file for a correct functioning of the tool. In the Terminal type the following commands:

```
cd /opt/lampp/etc
sudo chmod a+rw php.ini
nano php.ini
```

8.7. Search the following parameters in the sublime window that opened and replace them with the values given below:

```
post_max_size = 21000M
upload_max_filesize = 20000M
max_execution_time = 5000
max_input_time = 5000
memory_limit = 20000M
```

8.8. In the Terminal type the following command:

```
php /opt/lampp/htdocs/isomirwindow/artisan serve
```

8.9. Now the IsomiR Window can be accessed by typing http://127.0.0.1:8000 into the browser.

9. Export of the VM

VirtualBox can import and export virtual machines in the standard Open Virtualization Format (OVF). The Open Virtualization Format is uniform across a wide range of virtualization software products, which allows for virtual machines to be imported to a program like VirtualBox.

9.1. To export the Isomir Window VM in OVF format, go to the Virtualbox panel, click on the VM and select the File tab in the left upper corner of the window.
9.2. Choose the option that says **Export Appliance…**
9.3. When the Appliance Export Wizard appears, select the Isomir Window VM, click Next and then click Export.
9.4. The Isomir Window VM is ready to be imported, if desired, into a virtualization software product, namely Virtualbox.

Appendix C – Table of tools for isomiRs analysis

Tools	Annotation Analysis										Functional Analysis		
	Length distribu-tion of reads	Genome map	miRNA detection	Prediction of novel miRNAs	Detection of other non-coding RNA	miRNA editing detection	miRNA modification detection	miRNA SNP detection	Detection of Differentiall y expressed miRNA	Detection of Differentiall y expressed non-coding RNAs	Prediction of miRNA targets	Pathway analysis for miRNA targets	(GO) terms
miRanalyzer [7]			x	x			x		x				
SeqBuster [11]			x			x	x				x		
CPSS [8]	x	x	x	x	x	x	x		x	x	x	x	x
mirGator [9]		x	x	x			x		x		x	x	x
YM500 [10]			x	x					x				
isomiRex [19]			x				x		x				
miRspring [12]			x						x				
isomiRid [13]			x			x	x						
isomiRage [14]	x		x				x		x				
miRMOD [15]			x	x			x				x		
miR-isomiRExp [16]		x	x						x				
isomiR-SEA [17]			x				x	x	x		x		
DeAnnIso [18]	x	x	x			x	x	x	x		x	x	x